Circuit and CAD Solutions for Optimal SRAM Design in Nanoscale CMOS

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In partial fulfillment

of the requirements for the degree of

Doctor of Philosophy in Computer Engineering

by

Satyanand Nalam

December 2011

Approval Sheet

The dissertation is submitted in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy in Computer Engineering

_____
Satyanand Nalam (AUTHOR)

This dissertation has been read and approved by the examining committee

_____
Dr. Benton Calhoun (Advisor)

_____
Dr. John Lach (Committee chair)

_____
Dr. Mircea Stan (Committee member)

_____
Dr. Joanne Dugan (Committee member)

_____
Dr. Vikas Chandra (Committee member)

Accepted for the School of Engineering and Applied Science:

_____
Dean, School of Engineering
and Applied Science

(December, 2011)

Karmanye Vadhikaraste, Ma phaleshou kada chana,

Ma Karma Phala Hetur Bhurmatey Sangostva Akarmani—

"You have a right to perform your prescribed duty, but you are not entitled to the fruits of actions. Never consider yourself the cause of the results of your activities, and never be attached to not doing your duty."

– The Bhagavad Gita

Research advisor
Benton H. Calhoun

Author
Satyanand Nalam

Circuit and CAD Solutions for Optimal SRAM Design in Nanoscale CMOS

Abstract

Conventional 6T SRAM design involves balancing trade-offs among several critical metrics - yield, power, performance and area. Worsening variation makes scaling the 6T bitcell to newer technologies and lower supply voltage difficult, especially due to the need to balance the trade-offs between various metrics. In particular, lowering the SRAM voltage for low-power operation, while maintaining functionality, is a challenge. Several solutions have been proposed to ensure continued scaling of SRAM in cutting edge technology nodes and to keep lowering the minimum SRAM operating voltage (V_{MIN}). These include circuit solutions such as alternative bitcells and read/write assist techniques, technology solutions such as new materials and devices, and architectural solutions such as ECC and redundancy. Exploring this vast design space to zero in on an optimal design that is most suited to a designer's requirements thus becomes challenging. This dissertation makes the following three contributions to ensure continued scaling of SRAM to deep nanoscale technology nodes and to lower voltages.

First, we present five and six transistor bitcells that use asymmetric sizing of the cross-coupled inverter to improve read stability. Further, by removing the restriction of bitcell symmetry that makes balancing trade-offs in the conventional 6T bitcell challenging, these bitcells can use sizing as an effective knob to improve stability and to trade-off leakage power, read performance, writability, and area. An improvement in cell stability ensures scalability of SRAM to lower voltages for lower power, while maintaining acceptable levels of functional yield.

Second, due to the DC assumptions of conventional static read and write metrics, they are either optimistic or pessimistic in predicting cell failure for high performance SRAMs. The static metrics predict cell failure only by considering variation, while there are several other factors involved in the dynamics of a write operation that can cause it to fail. So, we define Dynamic write-limited V_{MIN} (DWV_{MIN}) for an SRAM that is based on $T_{WL-CRIT}$, a dynamic writability metric. DWV_{MIN} takes into account several other factors that can cause write failure and is a more accurate value of the lowest operating voltage for write-limited SRAMs.

Finally, the burgeoning SRAM design space has led to a designer productivity crisis. Thus, to improve productivity and enable a rapid and early exploration of the design space, we propose the Virtual Prototyping tool (ViPro). For any technology, ViPro produces an optimal base-case prototype of the SRAM, metric trade-off curves, and breakdown among various components. The designer can then iteratively explore the design space to reach an optimal final design. The Technology Agnostic Simulation Environment (TASE) component of ViPro can in fact be used as a stand-alone tool to port circuit analysis across technologies. This makes it a useful tool for any kind of circuit design.

Acknowledgments

First off, I am deeply grateful to my advisor, Dr. Ben Calhoun, for his guidance and support throughout my time in graduate school. He has played a major role in helping me mature as a researcher and set a great example with his dedication and work ethic. I am thankful to him for his faith in my abilities.

I had the most fortunate experience of working closely with Dr. Jiajing Wang, Dr. Randy Mann, Sudhanshu Khanna, and Jim Boley in the Robust Low Power VLSI SRAM group, and greatly benefited from their knowledge, experience, and the many exciting discussions we have had. A special mention goes to Jiajing for all her help during my early days in grad school, when I was relatively new to the field.

I would like to thank Dr. Vikas Chandra for mentoring me during my internships at ARM, and for his discussions, ideas, and help with various parts of this dissertation. I would also like to acknowledge Mudit Bhargava, Mark McCartney, and Dr. Ken Mai from Carnegie Mellon University, and Alexander Hoefler from Freescale Semiconductor for their insights and help during tapeouts. I am also grateful to the other members of my PhD committee, Dr. John Lach, Dr. Mircea Stan, and Dr. Joanne Dugan, for their time and suggestions for this work.

Thanks also go to all my other colleagues in the RLPVLSI group - Liang Di, Joe Ryan, Kyle Ringgenberg, Steve Jocke, Yousef Shakhsheer, Kyle Craig, Taeyoung Kim, Yanqing Zhang, Aatmesh Shrivastava, and Alicia Klinefelter for their suggestions and assistance over the years.

Also deserving of a mention are my friends and my house mate Saurav Basu, who have all made my time at UVA most memorable. Last but not the least, this milestone of my life would not have been possible without the love and support of my parents and sister.

Contents

Title Page i
Approval Sheet ii
Abstract iv
Acknowledgments vi
Table of Contents vii
List of Figures xi
List of Tables xvi
List of Acronyms xvii

1 Introduction **1**
1.1 Background and Motivation 1
1.1.1 The Power vs. Functionality Trade-off 2
1.1.2 Estimating V_{MIN} 4
1.1.3 SRAM Design Portability to Scaled Technologies 4
1.2 Major Contributions 7
1.2.1 Alternative Bitcells 7
1.2.2 Dynamic Write-limited V_{MIN} 8
1.2.3 SRAM Design Automation Tools 8
1.3 Organization 9

2 Five Transistor SRAM Bitcell **10**
2.1 Motivation 10
2.2 Earlier 5T bitcells 12
2.3 5T Bitcell Overview 13

2.3.1 Asymmetric Sizing 14
2.3.2 Read and Write Operations 15
2.3.3 5T Layout Topologies 17
2.4 Comparison with 6T 21
2.4.1 Read margin 22
2.4.2 Read current 24
2.4.3 Leakage 25
2.4.4 Hold margin 26
2.4.5 Write margin 28
2.5 Comparison with 8T 29
2.6 45nm Test Chip 33
2.6.1 Description of Structures 33
2.6.2 Measurements 34
2.7 Conclusions 37

3 Asymmetric Six Transistor Bitcell 39
3.1 Motivation 39
3.2 Related Work 39
3.3 Dual WL Asymmetric 6T Bitcell 41
3.3.1 Improving WNM and RSNM 41
3.3.2 Routing the Reset Signals 44
3.3.3 Sizing for Iso-area 45
3.4 Comparison with 6T 46
3.4.1 Noise Margins and V_{MIN} 47
3.4.2 I_{READ} 48
3.4.3 Half-select Stability 49
3.4.4 Bitcell Leakage 50
3.5 Comparison with 8T 51
3.6 Test Chip 53
3.7 Conclusions 53

4 Pseudo-differential Sensing for Single-ended Bitcells 55
4.1 Motivation and Related Work . . . 55
4.2 Pseudo-differential sensing . . . 56
4.2.1 Overview . . . 56
4.2.2 Pros and Cons . . . 58
4.3 Impact of Variation . . . 59
4.4 Comparison with Differential Sensing . . . 61
4.4.1 Read Delay . . . 61
4.4.2 Read Power . . . 61
4.5 Conclusions . . . 62

5 Dynamic Write-limited Minimum Operating Voltage for SRAM 64
5.1 Motivation and Related Work . . . 64
5.2 Critical WL Pulse-width . . . 66
5.3 DNM and DWV_{MIN} . . . 67
5.4 Factors affecting DWV_{MIN} . . . 71
5.4.1 WL pulse characteristics . . . 71
5.4.2 Memory size . . . 72
5.4.3 Bitcell parasitics . . . 73
5.4.4 Number of Cycles Prior to First Read . . . 76
5.5 Comparison with Static V_{MIN} . . . 78
5.6 Impact of Write Assists . . . 80
5.7 Conclusions . . . 81

6 Virtual Prototyping Tool 83
6.1 Motivation . . . 83
6.2 Related Work . . . 85
6.3 Overview . . . 87
6.3.1 Components . . . 89
6.3.2 Design Methodology . . . 90
6.4 TASE . . . 91

6.4.1 Tool Overview 92
6.4.2 Tool flow 95
6.4.3 Usage 97
6.5 Bitcell Generator 98
6.6 SRAM Model 99
6.6.1 Model Verification 102
6.7 SRAM optimization 105
6.8 Technology Agnostic SRAM Compiler 106
6.8.1 Schematic Generator 106
6.8.2 Layout Generator 107
6.9 Usage Examples 109
6.9.1 Base-case Generation 110
6.9.2 Re-optimization due to Process Change 112
6.9.3 Design Space Exploration 113
6.10 Conclusions 115

7 Conclusion **117**
7.1 Summary of Contributions 117
7.1.1 Alternative Bitcells 117
7.1.2 Dynamic Write Limited V_{MIN} 118
7.1.3 SRAM Design Automation 119
7.2 Future Work 120
7.2.1 Alternative Bitcells 120
7.2.2 Dynamic Writability 121
7.2.3 SRAM Design Automation 122
7.3 Conclusion 123

A Publications related to this dissertation **125**

List of Figures

2.1 Previously proposed 5T bitcells (a) [26] (b) [27] (c) [28] 13
2.2 (a) Bitcell schematic (b) 5T Read Static Noise Margin (RSNM) butterfly curve with "6T-like" sizing (c) 5T RSNM butterfly curve with asymmetric sizing [29]. 14
2.3 Waveforms for V_{DDC} write assist [29]. 16
2.4 6T bitcell layout cartoon showing the limiting design rules. Green indicates active n+ or p+ regions, red indicates polysilicon, yellow indicates contacts, and pink indicates n-well. The dashed line shows the bitcell outline. 18
2.5 Cartoons showing 5T bitcell layout topologies with (a) Increased bitcell height (b) Same bitcell height as 6T. 20
2.6 Comparison of the mean, standard deviation, and worst case RSNM of the 5T and 6T from a 1000 point MC at 0.7 V. 23
2.7 I_{READ} comparison between the 6T and the various 5T bitcells at 0.7 V. . . . 25
2.8 Total standby leakage for 5Tc. The leakage when storing a '0' is different from that when storing a '1'. 26
2.9 Leakage comparison between the 6T and the various 5T bitcells at 0.7 V. . . 27
2.10 Comparison of the mean, standard deviation, and worst case HSNM of the 5T and 6T from a 1000 point MC at 0.7 V. 27
2.11 Comparison of the mean, standard deviation, and worst case WNM of the 5T and 6T from a 1000 point MC at 0.7 V. 28
2.12 8T bitcell schematic. 29
2.13 8T bitcell layout cartoon. 31

2.14 (a) I_{READ} and (b) RSNM comparison for iso-area 5T and 8T cells at 0.7 V from a 1000 MC simulation. 32
2.15 45 nm bulk CMOS 5T test chip [29] 34
2.16 Schematic of 4 kb 5T block with write assist implementation [29]. 35
2.17 Measured logic analyzer waveforms showing (a) write and (b) read operations at 1V. 36
2.18 Impact of asymmetric sizing on write assist effectiveness. Measurements from a 4 kb array for each sizing are shown. Bit error rate is the number of observed bit fails divided by the size of the array. 37
2.19 Measurements from a 4kb bank showing the use of multiple write assists to improve writability. 37
2.20 Measured bit errors for 4 kb 6T and 5T banks with write assist across voltage. 38

3.1 5T with reset transistor [40]. 40
3.2 Proposed cell reset scheme. 42
3.3 Timing diagram for cell reset and write. 42
3.4 Options to share RSTG and RSTS for the reset transistor (a) column-wise and row-wise (b) row-wise and column-wise respectively. 43
3.5 Layout cartoon for asymmetric 6T with RSTG and RSTS routed column-wise and row-wise respectively. 44
3.6 Asymmetric 6T schematic. 45
3.7 Layout cartoons with normalized device widths for (a) conventional (b) asymmetric 6T bitcells. 46
3.8 Normalized Mean (a) RSNM and (b) WNM. The circular and the square markers represent the asymmetric cell and the conventional 6T respectively. 47
3.9 Normalized σ/μ of WNM at (a) 1 V and (b) 0.7 V at various process corners and voltages from a 1000 point MC simulation. The circular and the square markers represent the asymmetric cell and the conventional 6T respectively. The solid and dashed lines represent the write ‘0 and ‘1 cases respectively. . 48
3.10 Half-selected cell for (a) single WL and (b) dual WL asymmetric 6T. 50

3.11 Half-select cell RSNM during read for conventional 6T, and asymmetric 6T at 1V, TT, and $27^{\circ}\mathrm{C}$ from a 1000 point MC simulation. 51
3.12 Standby cell leakage for conventional and asymmetric 6T. 52

4.1 Column schematics for (a) 5T with pseudo-differential sensing and (b) 6T with differential sensing. 57
4.2 Bitline discharge cartoon for pseudo-differential sensing. ΔV1 and ΔV0 represent the BL voltage differential developed for reading '1' and '0' respectively. 58
4.3 SA input distributions for (a) Read '0' and (b) Read '1'. The distributions with the circular and cross markers are associated with the sensed and reference BL voltages respecitvely. 60
4.4 Read delay comparison for 5Te with pseudo-differential sensing and 6T with differential sensing at 1V. 62
4.5 Read power per column for 5Te and 6T at 1V is similar. 63

5.1 Transient state trajectories and the separatrix in the state-space demonstrating the relation between the $\mathrm{T_{WL-CRIT}}$ and the dynamic writability of the bitcell. For $\mathrm{T_{WL}} \geq \mathrm{T_{WL-CRIT}}$, the state changes and the write is successful. 67
5.2 A dynamically write limited but statically non-limited cell (a) becomes statically limited (b) as the voltage is lowered from 0.686V to 0.55V. 69
5.3 Using worst case $\mathrm{T_{WL-CRIT}}$ and $\mathrm{T_{WL}}$ to determine the dynamic writability limited $\mathrm{V_{MIN}}$. The intersection determines $\mathrm{DWV_{MIN}}$, 624 mV in this case. . 70
5.4 $\mathrm{DWV_{MIN}}$ dependence on voltage scaling of $\mathrm{T_{WL}}$. $\mathrm{DWV_{MIN}}$ increases from 624 mV (a) to 741 mV (b) for two different $\mathrm{T_{WL}}$ scaling approaches. 72
5.5 Impact of variability on $\mathrm{T_{WL-CRIT}}$ for different array sizes. 73
5.6 $\mathrm{DWV_{MIN}}$ dependence on array capacity. $\mathrm{DWV_{MIN}}$ increases from 624 mV for a 1kb array to 714 mV for a 5kb array. 74
5.7 $\mathrm{DWV_{MIN}}$ for various array sizes using SB. 74
5.8 Dominant bitcell parasitics. 75

5.9 Impact of inter-storage node parasitic on $T_{WL-CRIT}$ for each of the three most dominant capacitances, with the others kept constant. 75
5.10 DWV_{MIN} dependence on the bitcell parasitics. 76
5.11 Effect of the no. of cycles elapsed before the first read. 78
5.12 DWV_{MIN} dependence on the number of cycles prior to first read. DWV_{MIN} for a 1kb array lies between 624 mV and 744 mV. 78
5.13 Dynamic vs. Static V_{MIN} for self-timing path generated, heavily margined T_{WL} (a) and aggressive bitline differential dependent T_{WL} (b). 79
5.14 Impact of write assists on worst case $T_{WL-CRIT}$. Static write failure occurs without assist at 670 mV. No static write failures are observed above 500 mV with either assist. 80
5.15 DWV_{MIN} for various array sizes with and without write assists. Static $V_{MIN} < DWV_{MIN}$ for both assist methods. 81

6.1 Although there is some overlap between the functionality of ViPro and existing SRAM design tools, the novelty of ViPro is that it is the first such tool that fills the gaps between architectural simulators, transistor optimizers, and memory compilers. 87
6.2 Structure of ViPro showing various components, and inputs and outputs for the tool. 88
6.3 Methodology of using virtual prototypes for SRAM design. 92
6.4 Example (a) technology agnostic template (b) technology specific configuration. 94
6.5 Example execution file for TASE. 96
6.6 Tool flow diagram for TASE. 97
6.7 Optimal bitcell design through search space reduction. 99
6.8 SRAM architecture and hierarchy assumed by ViPro. 100
6.9 (a) SRAM parent class and (b) example component class showing circuit attributes and FoM estimation. 102
6.10 Hierarchical block diagram of the generated SRAM schematic. 107

6.11 Semi-automated layout for (a) 512x16 with column-mux of 1, (b) 64x128 with column-mux of 8. The annotations 1,2, and 3 refer to the bitcell array, WL drivers, and the bitslice leaf nodes for the two macros. 109
6.12 Generated SRAM schematic in the "new" 130 nm technology. The blocks starting from top-left are the WL Driver, bitcell array, WL buffer chain, bitslice, timing/predecode and predecode buffer chain. 111
6.13 Optimal E-D curve for generated base-case prototype using 45nm PTMs and re-generated basecase E-D curve after process model change. Memory capacity = 16kb, word-size = 16 bits. 112
6.14 Optimal base-case E-D curves for 65nm and 45nm PTMs generated by ViPro. Memory capacity = 16 kb, word-size = 16 bits. 114
6.15 Optimal E-D curve for SRAM prototype in PTM 45nm after changes in the bitcell and SA circuits. Memory capacity = 16kb, word-size = 16 bits. . . . 115
6.16 Current state of the work and potential future enhancements for (a) TASE (b) ViPro. 116

List of Tables

2.1 Normalized reference 6T sizing 18
2.2 Normalized limiting 6T bitcell design rules in a 45 nm technology 19
2.3 Normalized bitcell sizing. All cells same area as 6T. 22
2.4 Normalized 8T sizing 30

5.1 V_T offsets for static and dynamic write fails 70

6.1 SRAM energy verification with a 512x16 macro 103
6.2 SRAM delay verification with a 512x16 macro 104

List of Acronyms

5T	five transistor
6T	six transistor
7T	seven transistor
8T	eight transistor
10T	ten transistor
BL	bitline
BIST	Built-In Self-Test
CDF	Cumulative Distribution Function
CMOS	Complementary MOSFET
DIBL	Drain Induced Barrier Lowering
DNM	Dynamic Noise Margin
DRV	Data Retention Voltage
ECC	Error-Correcting Code
FinFET	Fin Field Effect Transistor
GPD	Generalized Pareto Distribution
HSNM	Hold SNM

HVT High Threshold Voltage

IC Integrated Circuit

IS Importance Sampling

LVT Low Threshold Voltage

MC Monte Carlo

MOSFET Metal-Oxide-Semiconductor Field Effect Transistor

MUX Multiplexer

NBTI Negative Bias Temperature Instability

NMOS N-type MOSFET

PBTI Positive Bias Temperature Instability

PMOS P-type MOSFET

PTM Predictive Technology Model

PVT Process, Voltage, and Temperature

RDF Random Dopant Fluctuation

RSNM Read SNM

SA Sense Amplifier

SB Statistical Blockade

SNM Static Noise Margin

SoC System-on-Chip

SOI Silicon on Insulator

SRAM Static Random Accessed Memory

VTC Voltage Transfer Characteristic

WL wordline

WSNM Write SNM

Chapter 1

Introduction

1.1 Background and Motivation

The scaling of integrated circuit technology over the past several decades has been predicted and driven by Moore's Law [1], which predicted that the number of transistors on a chip would continue to double every 18 months. Although technology scaling and the associated exponential rise in transistor count has contributed significantly to the improvement in density and performance of ICs, the complexity of the designs has increased a commensurate amount. In turn, this increased design complexity has been exacerbated by a more significant impact of previously second-order effects that have made IC design and manufacturing profoundly more difficult. The impact of these effects, such as gate-oxide tunneling, sub-threshold leakage current, drain-induced-barrier-lowering, process variation, manufacturing variation, and lithographic limitations, also seem to increase as technology scales down to the so called "end-of-the-roadmap" fundamental scaling limit.

Static Random Access Memory (SRAM) is a critical component of ICs, and is expected

to occupy over 90% of the chip area by 2013 [2]. Due to the comparitively higher density and smaller devices sizes of the SRAM bitcells, the scaling effects make SRAM design even more challenging than logic circuit design. Moreover, being the largest component in many embedded digital systems or Systems-on-Chips (SoCs) , SRAM power consumption dominates the overall power of the system, both in standby and operational modes. Thus, SRAM power reduction has become an increasingly important problem. Scaling down the supply voltage of the SRAM is a popular solution for power reduction. However, there are several challenges facing low-voltage/low-power SRAM design. The following subsections discuss these issues in detail and elaborate how this dissertation deals with the challenges of optimal SRAM design in nanoscale technologies.

1.1.1 The Power vs. Functionality Trade-off

Scaling down the supply voltage of the SRAM reduces the dynamic power, as well as both sub-threshold and gate leakage. Typically, an SRAM is designed to have sufficient hold, read, and write stability (usually quantified by static noise margins) as well as access speed under the nominal supply voltage. However, voltage scaling degrades these margins and reduces the functional yield. Thus there is a trade-off between yield and the power dissipation or energy consumption of the SRAM. This trade-off can be quantified by the minimum supply voltage (V_{MIN}) at which the yield is acceptable.

Another important factor that affects the functional yield and consequently the V_{MIN} of the SRAM is variation. Based on the scale at which it occurs, variation can be classified into two categories – local and global. On one hand, global variations occur on the die-to-die scale and systematically or uniformly influence all the transistors on the same

die. They mainly include the inter-die manufacture related *process* variations and environmental conditions including *voltage* supply fluctuations and *temperature* change (e.g. PVT variations [3]). On the other hand, local variations such as threshold voltage (V_T) variation due to random dopant fluctuation (RDF) occur within a die and cause mismatch between adjacent devices.

SRAM V_{MIN} is extremely sensitive to local variation because of three reasons. First, to achieve higher density, the SRAM cell uses devices with much smaller geometry (e.g. width and length) than regular logic gates. RDF induced random V_T variation is normally distributed with a standard deviation (σ) inversely proportional to the square root of the transistor channel area [4]. Thus, SRAM bitcell transistors have random V_T variation with a larger value of σ, consequently increasing the σ of the stability metrics/noise margins as well. Second, SRAM capacities have been scaling up from few hundered KB to a few GB due to the higher density facilitated by technology scaling. The higher number of samples makes it quite likely for large variations in SRAM noise margins beyond 6σ to occur. Finally, many SRAM metrics, stability metrics in particular, are susceptible to mismatch because the SRAM cell typically uses two symmetrical cross-coupled inverters. A small mismatch between adjacent transistors within the two inverters can lead to a large variation in the cell's behavior.

Clearly, improving the cell stability, as well as reducing the impact of variation would help lower the V_{MIN}, resulting in lower power while maintaining an acceptable functional yield. In this dissertation, we propose two alternative bitcells that aim to improve the read/write stability and reduce the variation impact so that lower V_{MIN} can be achieved for SRAMs in scaled technologies.

1.1.2 Estimating V_{MIN}

As discussed in the previous section, improving the cell stability and reducing the impact of variation enables lower V_{MIN} and better power savings. However, estimating the impact of these stability improvements on SRAM V_{MIN} is equally important. An underestimation of V_{MIN} causes unacceptable failures while an overestimation results in higher energy or power consumption. V_{MIN} estimation based on conventional static noise margin metrics has been proposed [5]. However, static metrics are either pessimistic or optimistic due to their assumption of DC operating conditions for their evaluation. This is because the impact of noise on cell stability depends not only on the amplitude of the noise, but also its duration. The error caused by using static metrics for V_{MIN} prediction can be expected to be more pronounced as the performance of the memory increases and access times shrink.

Dynamic metrics have been proposed [6][7][8][9] to replace static noise margin metrics as more accurate measures of cell stability. However, predicting the SRAM V_{MIN} based on dynamic metrics is as yet unexplored. In this dissertation, we investigate dynamic metrics for SRAM stability and describe how they can be used to predict the SRAM V_{MIN} more accurately.

1.1.3 SRAM Design Portability to Scaled Technologies

Researchers have proposed several solutions to combat SRAM design challenges, so that lower V_{MIN} can be achieved. These techniques span the entire range of design abstraction, starting from fundamental device solutions, to circuit and architectural methods. For example, at the device level, new technologies such as multi-gate transistors, carbon nanotubes, and organic molecular transistors are emerging [10]. An SRAM bitcell based on

FinFET transistors can offer higher read stability and writability than a conventional planar SRAM cell due to lower V_T variation [11]. However, these new technologies and devices are not yet mature and it is quite likely that CMOS technology will continue to be used in the foreseeable future. Thus, it is important to overcome the scaling challenges of CMOS technology for SRAM design.

The combination of density, performance, and compatibility with the CMOS logic process has made it hard to dislodge the 6T SRAM bitcell as the primary memory element even in deep nano-scale technologie (e.g. beyond the 45 nm node). To maintain sufficient stability at lower voltage in the face of increasing variation effects, circuit solutions such as voltage-bias based read and write assist methods have been proposed [12][13][14]. These methods fall broadly into two categories – ones that impact the strength of the cross-coupled inverters in the bitcell and ones that affect the strength of the access transistors [15]. A fundamental limitation of the 6T bitcell is that the requirements imposed on the bitcell (e.g. in terms of device sizes) are contradictory for read and write operations. This makes it difficult to improve both read and write noise margins simultaneously. Thus, an alternative approach is to change the structure of the bitcell so that this restriction is removed, enabling a simultaneous improvement in both functionalities, and consequently a lower V_{MIN} [16][17][18].

Instead of improving the devices and circuits in an SRAM so that failures are minimized and yield is acceptable at low voltage, an alternative approach is to allow the failures to occur and fix them. Architectural approaches such as using redundant rows, columns or blocks in an SRAM or using Error Correction Codes (ECC) are an example of such solutions to SRAM scaling challenges.

This plethora of techniques to address SRAM scaling challenges such as leakage and variation has led to a considerably expanded design space. Searching for the optimum solution to design even one SRAM is becoming increasingly complex. Further, in today's IC industry, the growing number of transistors that can now be integrated on a single die has resulted in an increase in the number of SRAMs that are embedded onto a single die (e.g. as caches). Traditionally, porting designs from a previous technology was straightforward. However, due to the problematic effects of the deep nanoscale processes, many previously used designs and techniques are no longer viable and designs often need to start from scratch every generation. All these factors have resulted in a designer productivity and design time crisis.

While one solution is to increase the size of the design teams and have the designers work at higher levels of abstraction (e.g. manipulating blocks of pre-designed logic gates rather than individual transistors), this is not sustainable for two reasons. One, it is not possible for the team size to keep pace with the exponentially increasing transistor count. Second, increasing the level of design abstraction is difficult in the era of nanoscale process technologies because the underlying physics of the semiconductors are causing increasing problems that propagate to the upper levels of the abstraction hierarchy.

In this dissertation, we develop design automation tools and methodologies for SRAM design. These tools enable a rapid exploration of the burgeoning SRAM design space and ease design porting to new technologies, thus enhancing designer productivity.

1.2 Major Contributions

In this thesis, we mainly address the challenges involved in designing an optimal low-power SRAM in deep nanoscale technologies. First, we present circuit solutions such as new alternative SRAM bitcells that increase the functinoal margins making it possible to reduce the V_{MIN}. Second, we present a new methodology that can help estimate the V_{MIN} with greater accuracy. Third, we present new CAD tools that improve SRAM designer productivity and facilitate easier exploration of the design space consisting of several such solutions for lowering SRAM V_{MIN} (e.g. alternative bitcells, assist methods, process technology improvements etc.). These contributions are elaborated in the following sub-sections.

1.2.1 Alternative Bitcells

We present a 5-transistor and an asymmetric 6-transistor SRAM bitcell. Both bitcells are based on the concept of asymmetrically sizing the cross-coupled inverters in the bitcell. We show how sizing can be used as a knob to improve read stability in the 5T, and both read stability and writability in the asymmetric 6T, while trading-off other metrics such as performance, area, and leakage as per the design requirements. We present measurement results from a 45 nm test-chip that demonstrate the functionality of the 5T SRAM. Finally, since both the bitcells are single-ended, we present a single-ended sensing scheme for our bitcells.

1.2.2 Dynamic Write-limited V_{MIN}

We first introduce $T_{WL-CRIT}$, a measure of dynamic writability that takes into account the impact of factors not considered by static writability metrics. For instance, the WL pulse width, the parasitic capacitances in the bitcell, and the time of occurence of the first read after a write, all influence the dynamic writability of the cell, and consequently the V_{MIN} of the cell. We demonstrate how $T_{WL-CRIT}$ can be used to estimate the Dynamic write-limited V_{MIN} (DWV_{MIN}) of an SRAM. We then investigate the impact of these dynamic factors on the DWV_{MIN} and show how they cause the actual V_{MIN} of the SRAM to be different from that predicted in a static manner.

1.2.3 SRAM Design Automation Tools

We present two tools that help automate SRAM design and enhance productivity. Technology Agnostic Simulation Environment (TASE) helps port groups of simulations that are tied to the analysis of a particular design from one technology to another. It can be used to ease circuit simulations for any kind of digital or analog circuit design. The Virtual Prototyping tool (ViPro) generates a base-case starting point SRAM in a new technology and provides trade-off curves between SRAM metrics such as access time and the energy per access. ViPro uses TASE to characterize the component circuits of an SRAM to determine the optimal design parameters (e.g. device sizes) for the SRAM and uses these parameters to automatically or semi-automatically generate full schematic and layout for the SRAM.

1.3 Organization

This dissertation is constructed in the following manner. Following the motivation and background provided in this chapter, chapter 2 describes the first of our alternative bitcell designs, the five-transistor (5T) bitcell. In this chapter, we introduce the key idea of asymmetric sizing and how it affects SRAM trade-offs. Chapter 3 describes our second alternative bitcell design, the asymmetric 6T bitcell. This bitcell builds on the asymmeric sizing concept used in the 5T, and attempts to address the drawbacks of the 5T. Chapter 4 explores single-ended sensing for our proposed bitcells.

In chapter 5, we discuss dynamic writability metrics for SRAM, introduce the concept of a dynamic write-limited V_{MIN} (DWV_{MIN}) for SRAM, and understand the factors that can impact the DWV_{MIN}.

In chapter 6, we switch focus to productivity-enhancing SRAM design automation tools and methodologies. We discuss the Virtual Prototyper (ViPro) and the Technology Agnostic Simulation Environment (TASE) tools and the SRAM design methodology using these tools.

Finally, in chapter 7, we summarize the ideas and contributions presented in this thesis and discuss potential directions for future work.

Chapter 2

Five Transistor SRAM Bitcell

2.1 Motivation

Technology scaling is accompanied by shrinking transistor widths and lengths and a scaling down of supply voltage. This leads to lowering of noise margins in retention or hold, as well as during read and write operations. Moreover, as device dimensions shrink, the effect of inter-die and intra-die variations increases. In particular, shrinking widths and lengths of transistors increases the variability (e.g. standard deviation) of the distribution of threshold voltage (V_T) of the transistors in the bitcell. Consequently, the variability of noise margins increases [19]. This requires designers to look farther out into the tails of these distributions for the worst cases and design for these to meet yield requirements. Ultimately, this leads to conservative and pessimistic designs, which consume more area and power than necessary, for a majority of the chips produced. Thus, scaling the traditional six transistor (6T) bitcell to newer technology nodes and to lower voltages is difficult. In addition, the need to satisfy conflicting requirements on the bitcell for performance, area

and stability, and also for read and write operations, while maintaining symmetry, makes SRAM design even more challenging.

Alternative bitcells, such as the 8T, that decouple the read and write operations and eliminate some of the conflicting trade-offs have been proposed [16] and implemented commercially in processor caches [20][21][22][23]. Alternatively, designers have explored other options for improving 6T stability, such as read and write assists [14][12][24][25]. In either case, the ultimate goal is to improve bitcell stability so that lower power operation can be achieved by reducing the minimum operating voltage (V_{MIN}).

Both these solutions incur an area penalty, while read and write assist methods can potentially incur power and performance overheads as well. In particular, the 8T bitcell represents a drastic jump in the stability-area space since it improves the RSNM to be equal to the HSNM while adding a significant area overhead. Further, certain assist methods can increase the susceptibility of half-selected cells (e.g. cells whose WL is turned on but BLs are not selected) to disturbs. Thus, there is a need for a bitcell that can provide better trade-offs between area, stability, performance, and leakage and bridge the gap between the 6T and the 8T bitcells to enable a more optimal SRAM design.

In this chapter, we look at a five transistor (5T) bitcell that leverages asymmetric sizing of the cross-coupled inverters to improve the read stability, which is quantified by Read Static Noise Margin (RSNM). Further, we see how asymmetric sizing becomes a knob to trade-off area, read delay, and bitcell standby leakage due to the removal of the symmetry requirement.

2.2 Earlier 5T bitcells

Several 5T bitcells have been proposed earlier. In [26], the authors propose a 5T bitcell (Figure 2.1a) that focuses primarily on saving area and reducing leakage. They use asymmetric sizing, combined with a intermediate bitline (BL) precharge voltage to solve the single-ended write problem (discussed later in the chapter) at the cost of degraded RSNM. Further, this approach is sensitive to variation, which is not investigated in their work. The 5T bitcell proposed in this work primarily focuses on improving RSNM, while also using asymmetric sizing as a more generic method to trade-off other critical metrics. We also investigate the impact of variation through Monte Carlo simulations.

Tran presents a 5T bitcell in [27] which solves the single-ended write issue by using a write assist method that weakens the feedback of the cross-coupled inverters in the cell (Figure 2.1b). However, his method requires a long BLs with large number of bitcells per column in order to ensure the unselected cells are not susceptible to upsets. This in turn affects performance. In our proposed bitcell, asymmetric sizing can be used to improve RSNM without comprimising on the performance.

In [28], the authors present a portless 5T cell. This cell is fundamentally different from the ones described so far in that it does not have an access transistor. Instead, the storage nodes are accessed by the BLs through the PMOS pull-up devices. The sizing of the AXS transistor (Figure 2.1c) is used as a knob to trade-off performance, stability, and leakage power with bitcell area. In particular, to achieve similar performance in terms of read current (I_{READ}), the 5T bitcell area needs to be about 1.5$\times$ a conventional 6T bitcell. The proposed 5T bitcell with asymmetric sizing can provide better I_{READ} and reduced variability in I_{READ} for the same area as the 6T.

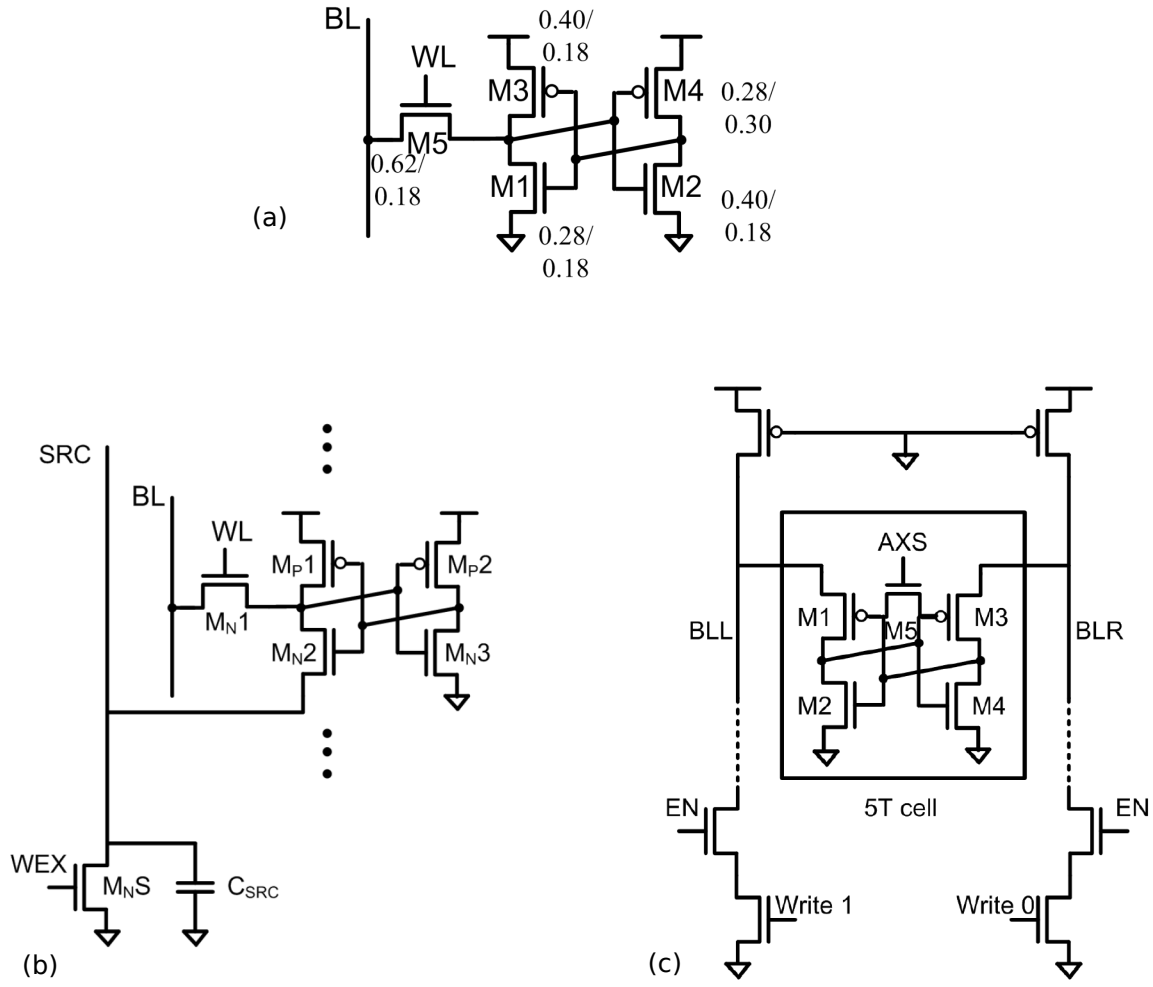


Figure 2.1: Previously proposed 5T bitcells (a) [26] (b) [27] (c) [28]

2.3 5T Bitcell Overview

In this section, we look at the key idea behind asymmetric sizing and describe the read and write operations of the 5T SRAM. We then analyze layout topologies for the 5T that can potentially leverage the area gained by using a lone access transistor. By using different asymmetric sizing approaches for the cross-coupled inverters, the designer can trade-off area, stability, read current (I_{READ}), and cell leakage in different ways.

2.3.1 Asymmetric Sizing

Fig. 2.2(a) shows the schematic of the proposed 5T bitcell [29], which is simply a 6T missing one access transistor. Read and write accesses are similar to the 6T, except that they are single ended through the lone access device. If '6T-like' sizing is used (e.g. W_{N1}=W_{N2}, W_{P1}=W_{P2}), the lower lobe of the resulting read butterfly curve is squashed due to the voltage-dividing effect of transistors N1-N3 (see Fig. 2.2(b)). The absence of the second access transistor results in the upper lobe being larger. This lobe is the same as in the unperturbed hold or retention state. The RSNM of the 5T bitcell is determined by the smaller lobe of the butterfly curve and equals the side of the square that can be embedded in this lobe [30]. Thus, in the nominal case, the 6T and the "6T-like" 5T have the same RSNM, determined by the embedded square in the smaller lower lobe of the butterfly curve.

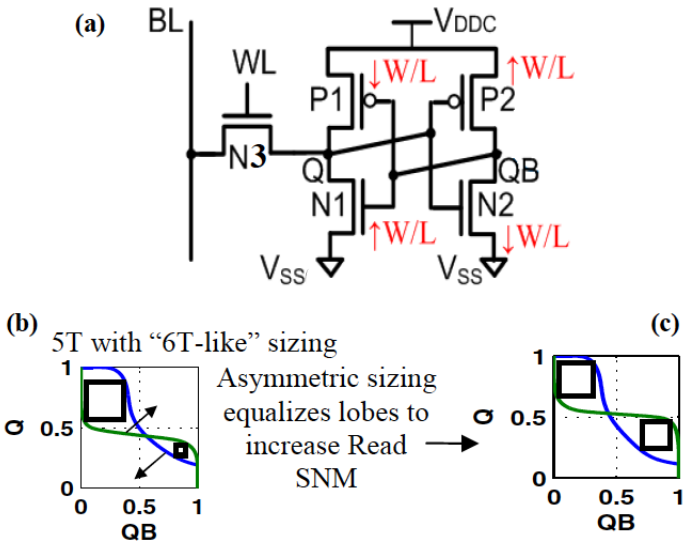


Figure 2.2: (a) Bitcell schematic (b) 5T Read Static Noise Margin (RSNM) butterfly curve with "6T-like" sizing (c) 5T RSNM butterfly curve with asymmetric sizing [29].

Now, if the voltage-transfer characteristics (VTCs) of the cross-coupled inverters can be modified as indicated by the arrows in Fig. 2.2(b), the two lobes become more similar in

size, as seen in Fig. 2.2(c), leading to an increase in the RSNM of the 5T when compared to the 6T or the "6T-like" 5T. The skewing of the individual inverter VTCs can be achieved either by sizing or by using different threshold voltages (V_T's). We use asymmetric sizing to achieve RSNM improvement since having multiple V_T's increases the cost of fabrication with each new V_T requiring an additional mask [31]. The cross-coupled inverters in the bitcell can be made asymmetric by changing the width to length (W/L) ratios as indicated in Fig. 2.2(a). Depending on the sizing approach used, stability, I_{READ}, and cell leakage can be traded off in different ways, as demonstrated by the examples in Section 2.4.

2.3.2 Read and Write Operations

Read

The 5T SRAM bitcell has a single-ended read operation through the lone access transistor, unlike the differential operation in the conventional 6T. This throws up new challenges pertaining to overall read performance and robustness in the presence of noise on the BL. Chapter 4 describes the pseudo-differential sensing scheme that is proposed for reading single-ended cells such as the 5T.

Write

In a conventional 6T, writing '0' and '1' are essentially the same operation. Both involve passing a low voltage (e.g. a '0') to the storage node of the cell that originally stores a high value (e.g. '1'). Since the NMOS pass-gate is stronger than the PMOS pull-up, this node gradually discharges. Once it crosses the trip-point of the other inverter, the PMOS pull-up on the opposite side turns on. Following that, the write completes quickly due to

the feedback of the inverters hastening the flipping of the cell.

Writing a '0' to the 5T bitcell is not an issue since the NMOS access transistor, N3, can pass a strong '0'. However, due to the absence of the second access transistor, writing a '1' is impossible without a circuit assist since there is no way to discharge the complementary node storing a '1'. N3 by itself cannot be used to complete the write operation since it cannot pass a strong '1'. Moreover, if the sizing approach used strengthens N1, writing a '1' becomes even more challenging due to the ratioed fight between N3 and N1. The asymmetric 6T bitcell we proposed in [32] and describe in Chapter 3 exploits the benefits of asymmetric sizing towards RSNM by giving up on area benefits, to avoid paying a penalty in terms of writability.

One method of solving the write issue for the 5T is to collapse V_{DDC} [12][24]. As the timing waveforms in Fig. 2.3 show, collapsing V_{DDC} weakens the cell feedback, enabling it to flip despite the weak '1' passed by N3. We restore V_{DDC} to full rail before the end of the WL pulse to ensure completion of the write operation.

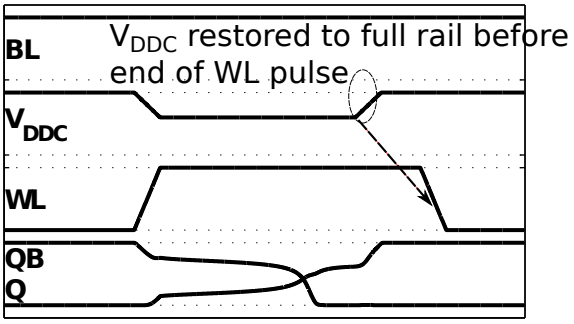


Figure 2.3: Waveforms for V_{DDC} write assist [29].

V_{DDC} can be routed row-wise or column-wise, the latter being more compatible with conventional bitcell layout. Collapsing V_{DDC} column-wise reduces the HSNM of the un-

selected cells on the same column (e.g. WL=0). On the other hand, a row-wise V_{DDC} scheme reduces the stability of the half-selected cells (e.g. WL=1, BLs precharged). Using a pulsed V_{DDC} scheme maintains a dynamic margin in the half-selected cells higher than static NM [33]. For row-wise routed V_{DDC}, we can either write the entire row at once (e.g. no interleaving of words in a row), or use a row-wise read-modify-write approach [13].

Other write assists can be used in conjunction with collapsing V_{DDC}, such as boosting the WL [34]. This allows for a smaller swing in V_{DDC} while writing, thus reducing the dynamic power consumption, particularly if the V_{DDC}s for entire columns are lowered. Section 2.4 demonstrates example 5T bitcells where the asymmetric sizing approach used affects the WNM in different ways. The measurements in section 2.6 further confirm the effectiveness of the write assists.

2.3.3 5T Layout Topologies

We start with the pushed-rule reference 6T SRAM bitcell (device sizes normalized to default device length in Table 2.1). For the 45 nm technology that we use, the cell dimensions and the area are determined by a subset of the pushed design rules, depicted in Fig. 2.4. These rules are listed in Table 2.2. The width (X_{6T}) and height (Y_{6T}) of the cell can be determined by equations (2.1) and (2.2) respectively. Plugging in the values of the design rules, we get X_{6T}=18.91, Y_{6T}=6.55, and a cell area A_{6T}=123.86, which agrees with the values as measured from the layout.

$$X_{6T} = 2.(\frac{1}{2}(TT) + GPA + max(W_{PD}, W_{PG}) + NP + W_{PU} + \frac{1}{2}(AA)) \qquad (2.1)$$

Table 2.1: Normalized reference 6T sizing

Device	**W/L**
Pull-up (W_{PU}/L_{PU})	1.27/1
Pull-down (W_{PD}/L_{PD})	4/1
Pass-gate (W_{PG}/L_{PG})	2.91/1.04

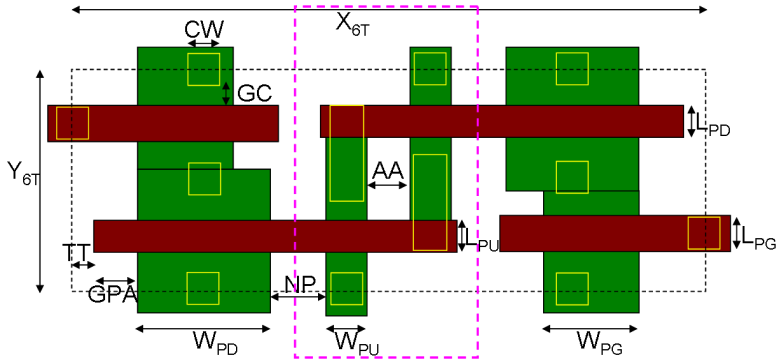


Figure 2.4: 6T bitcell layout cartoon showing the limiting design rules. Green indicates active n+ or p+ regions, red indicates polysilicon, yellow indicates contacts, and pink indicates n-well. The dashed line shows the bitcell outline.

$$Y_{6T} = 2(CW) + 4(GC) + max(L_{PD}, L_{PU}) + L_{PG} \tag{2.2}$$

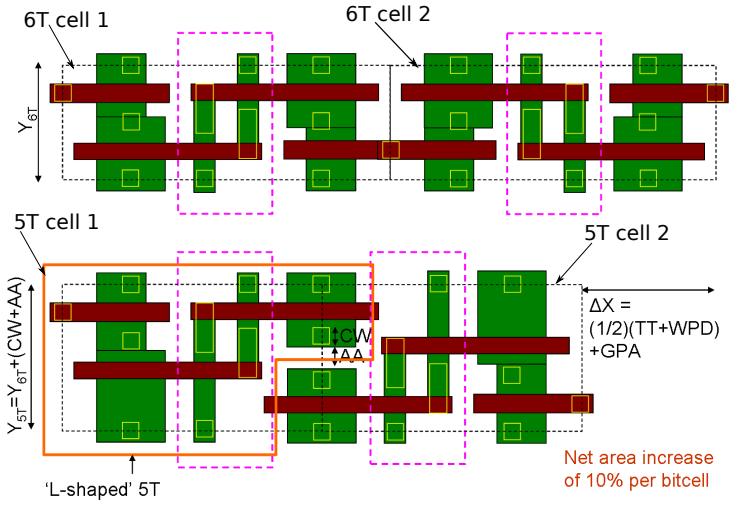
The 5T bitcell is L-shaped due to the missing access transistor. This space can potentially be exploited by rotating and mirroring the adjacent bitcell on the same row, so that the notches overlap and the cells fit together (Fig. 2.5(a)). The N2 transistors of the abutted cells can be aligned to reduce the effective width of the bitcell. We find that the cell width can be reduced by 28% compared to the 6T with this topology. However, this increases the height of the cell due to the 'AA' spacing necessary between the two n+ active regions and an additional contact. The short and wide aspect ratio of the cell implies that increasing

Table 2.2: Normalized limiting 6T bitcell design rules in a 45 nm technology

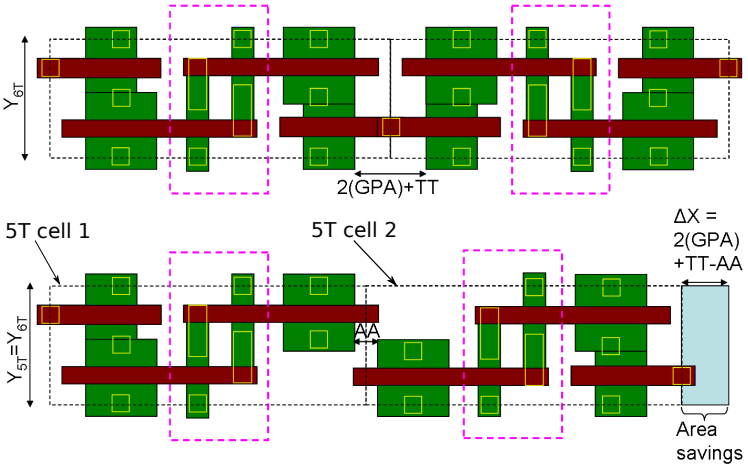
Design Rule	**Symbol**	**Value**
Gate to Contact space	GC	0.64
Gate past Active	GPA	1.27
Gate tip to tip	TT	1.31
Contact size	CW	0.98
p+ to p+ space	AA	1.24
n+ to p+ space	NP	1.64

the height of the cell would have a large impact on the cell area. For a "6T-like" 5T bitcell with no asymmetric sizing, we find that this topology results in a cell area of 135.66 (13.62x9.96), which is 10% larger than the reference 6T. Thus, we rule out this topology for the 5T bitcell.

Alternatively, the cell height can be kept unchanged and a small reduction in width (1.3 units) is obtained in the topology shown in Fig. 2.5(b). This reduction is due to the fact that the limiting rule on the side of the bitcell without the access transistor is only the 'AA' spacing rather than a combination of '$\frac{1}{2}$TT' and 'GPA'. The cell width for this topology is given by equation (2.3). We find that the "6T-like" 5T has a cell area of 117.64 (17.96x6.55), a saving of 5%. Further area gains can be achieved in an asymmetric 5T with



(a)



(b)

Figure 2.5: Cartoons showing 5T bitcell layout topologies with (a) Increased bitcell height (b) Same bitcell height as 6T.

a reduced W_{N2}.

$$X_{5T} = (\frac{1}{2}(TT) + GPA + max(W_{PD}, W_{PG}) + NP + W_{PU} + \frac{1}{2}(AA)) + (\frac{1}{2}(AA) + W_{PU} + NP + WPD + \frac{1}{2}(AA)) \quad (2.3)$$

Thus, by sizing the cross-coupled inverters the same as the 6T, the 5T can save area for the same RSNM and I_{READ} as the 6T. Alternatively, at iso-area, the 5T can have better RSNM and I_{READ} through asymmetric sizing, for example, by sizing up the N1 pull-down transistor. In general, depending on the asymmetric sizing approach used, the 5T can trade-off metrics such as I_{READ} and cell leakage in different ways, in addition to improving the RSNM, as demonstrated in section 2.4.

2.4 Comparison with 6T

The 5T improves RSNM relative to the 6T either by strengthening N1 or P2 relative to the reference 6T, or by weakening P1 or N2, or by a combination of these approaches. Apart from a definite increase in RSNM, asymmetric sizing also provides a knob to trade-off other metrics critical to SRAM, such as area, read current and leakage. The exact sizing approach determines how these other metrics are traded-off.

In this section, we use five asymmetric 5T cells with different sizing approaches (Table 2.3) to demonstrate how the 5T can trade-off various metrics in addition to improving RSNM, giving it the flexibility to target different application needs. These cells use sizing approaches that target the width, length, or both for each of the transistors in the cross-coupled inverters. All of these cells have the same bitcell area as the reference 6T cell. In 5Ta, N2 is weakened by making it minimum width. The area gained is leveraged to increase W_{N1}. The width gained due to the missing access transistor allows us to size N1 up further so that the area of the resulting asymmetric 5T is the same as the reference 6T. In 5Tb, the extra area gained is used to increase L_{P1} and W_{N1} to weaken P1 and strengthen N1 respectively. In 5Tc, P2 is strengthened by increasing its width. In 5Td, N2 is weakened by

both increasing its length and reducing its width, in addition to widening N1. Finally, 5Te is the same as 5Ta, with the exception of the access transistor N3, which is sized up. All these cells have the same area as the reference 6T. The following subsections demonstrate how the sizing approaches trade-off various metrics differently.

Table 2.3: Normalized bitcell sizing. All cells same area as 6T.

Bitcell	**P1**	**P2**	**N1**	**N2**	**N3**
5Ta	1.27/1	1.27/1	7.67/1	1.27/1	2.91/1.04
5Tb	1.27/1.35	1.27/1	6.73/1	4/1	2.91/1.04
5Tc	1.27/1	4.95/1	4/1	1.27/1	2.91/1.04
5Td	1.27/1	1.27/1	5.45/1	2.54/1.35	2.91/1.04
5Te	1.27/1	1.27/1	7.67/1	1.27/1	3.64/1.04

2.4.1 Read margin

Fig. 2.6 shows the mean, standard deviation (sigma) and worst case RSNM at 0.7 V from a 1000 point Monte Carlo (MC) with the sizing approaches listed in Table 2.3. We observe that all approaches improve the mean RSNM by different amounts, ranging from 31% for 5Td to 42% for 5Ta. Further, a sizing approach that strengthens the inverter P1-N1 leads to a better improvement in mean RSNM (5Ta,b). This is likely because strengthening P1-N1 has a more direct impact on the RSNM by strengthening the node that is susceptible to read disturb. All the sizing techniques also affect other metrics such as I_{READ}, leakage, write margin and variation tolerance in different ways, which we will see in the remainder of this section.

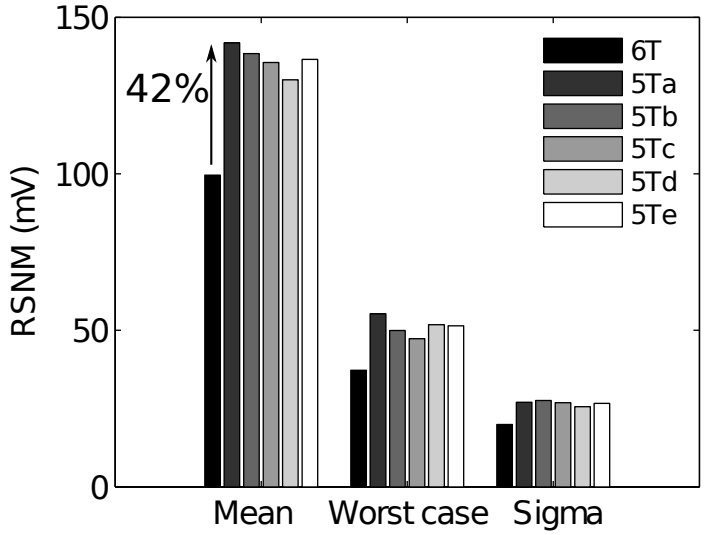


Figure 2.6: Comparison of the mean, standard deviation, and worst case RSNM of the 5T and 6T from a 1000 point MC at 0.7 V.

We also note that the sigma of the 5T RSNM distributions is larger than that of the 6T RSNM. This can be explained as follows. The 6T RSNM is the side of the embedded square in the smaller of the upper and lower lobes of the butterfly curve. Taking the minimum of the two values reduces the mean of the resulting distribution and squashes it, reducing its sigma. The distribution is no longer normal, but long tailed [35]. For the 5T case, one lobe remains much larger than the other (although asymmetric sizing reduces this gap). So, the RSNM always corresponds to a single lobe, the lower lobe in Fig 2.2, and remains normally distributed. Though, the sigma of the distribution is larger than that of the 6T due to this, we note that for a given bitcell variation, the RSNM butterfly curve lobe for the 5T is always bigger than the 6T. Thus, the RSNM of the 5T can be no worse than that of the 6T.

For example, the sigma of the upper and lower lobes for the 6T RSNM is 26 mV. The sigma of the minimum distribution is 20 mV. For 5Ta, the sigma of the larger and smaller

lobes are 22 mV and 27 mV respectively. The sigma of the RSNM corresponds to that of the smaller lobe, 27 mV. Though the sigma of the 5T distributions is a larger, we see from Fig. 2.6 that the 5T still has better worst case RSNM than the 6T for all the asymmetric sizing approaches.

Finally, the stability of the half-selected cells also improves due to asymmetric sizing. Moreover, the single-ended nature of the 5T means that there is no read or half-select stability issue when the data stored is a '1', whereas the half-select problem affects the 6T irrespective of the data stored.

2.4.2 Read current

A sizing approach that strengthens the read pull-down (N1) also helps increase the bitcell current during a read (I_{READ}). All the cells listed in Table 2.3 except 5Tc have a stronger N1 compared to the 6T resulting in higher I_{READ} (Fig. 2.7), and thus faster BL discharge during a read. Further, widening N1 also reduces the variability of I_{READ}, and all the bitcells except 5Tc have a worst case I_{READ} at least 20% larger than 6T.

Sizing up the access transistor is even more effective at improving I_{READ}, while not imposing any extra area penalty. For instance, 5Te has a mean I_{READ} that is 30% larger than the 6T, while the other cells improve the I_{READ} only by 4-6%. For the sizing that we use, this improvement comes at the expense of only a slightly reduced RSNM benefit in the worst case, as seen in Fig. 2.6. After a certain point, the RSNM penalty due to the upsized access transistor will nullify any improvements provided by asymmetric sizing.

Though asymmetric sizing in a 5T can speed up the BL discharge due to the increased I_{READ}, since the cell is single-ended, it is a challenge to translate this into an improvement

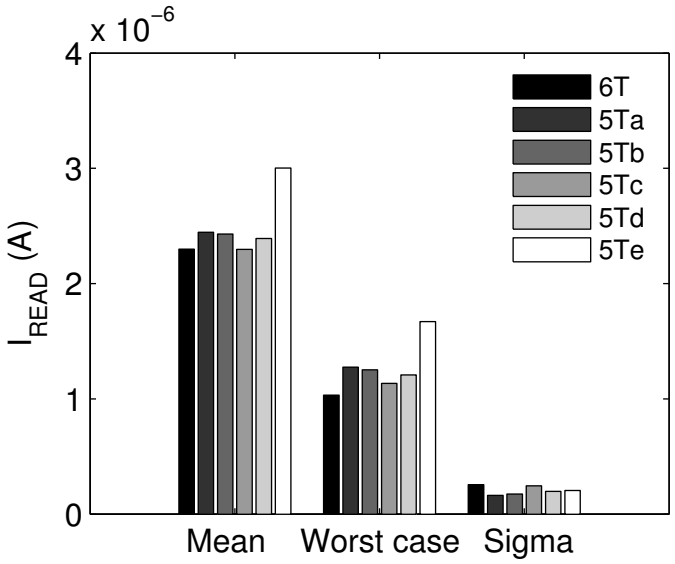


Figure 2.7: I_{READ} comparison between the 6T and the various 5T bitcells at 0.7 V.

in overall read performance that includes the sensing delay. This is because logic-gate based full-swing sensing that is conventionally used for single-ended cells such as the 8T, is slower than differential sensing [36]. We investigate sensing schemes for 5T in chapter 4.

2.4.3 Leakage

The leakage components are the leakage through the pull-up, pull-down, and access transistors (e.g. bitline leakage). We consider only the subthreshold leakage component since the gate leakage is negligible for this technology. The 5T bitcell has data dependent leakage due to asymmetric sizing of the cross-coupled inverters. Note that there is no bitline leakage when storing a '1'. Also the leakage through one pull-up (or pull-down) can be different from that through the other depending on the sizing used. Fig. 2.8 shows the cell leakage across V_{DD} for the 5Tc bitcell. We observe that the average leakage of the cell is lower than that of the 6T across the voltage range. For this 5T bitcell, the leakage through

P2 is 44% higher than that through P1 due to the larger width of P2.

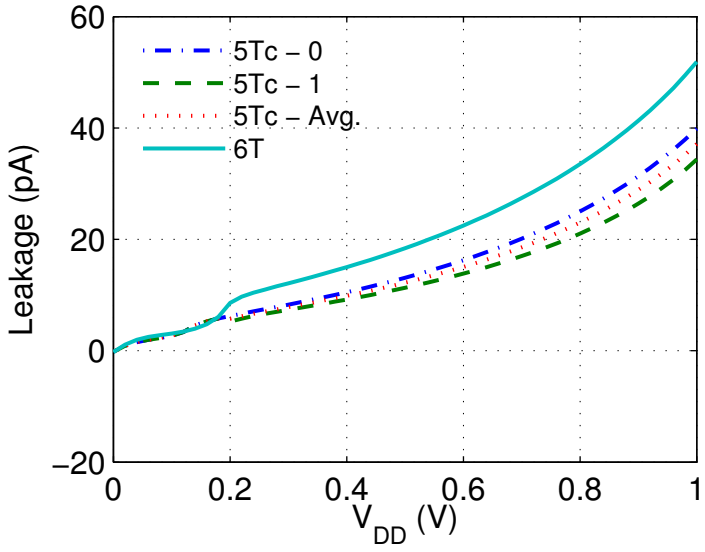


Figure 2.8: Total standby leakage for 5Tc. The leakage when storing a '0' is different from that when storing a '1'.

Fig. 2.9 plots the leakage for the 5T and 6T cells at 0.7 V. We observe that for different sizing approaches, the leakage is higher for different data. For example, the leakage when storing '1' is higher for 5Ta than when storing a '0', while the opposite occurs for 5Tc. However, for all sizing approaches, the 5T leakage is lower than the 6T for both '0' and '1'.

2.4.4 Hold margin

To improve RSNM, asymmetric sizing of the cross-coupled inverters attempts to equalize the size of the butterfly curve lobes during read, as observed in Fig. 2.2. However, this skews the initially symmetric lobes of the curve during hold. Consequently, the HSNM of the 5T bitcell, as measured by the Seevinck method, degrades when compared to the 6T. However, as observed in Fig. 2.10, this degradation is minimal, with the mean HSNM

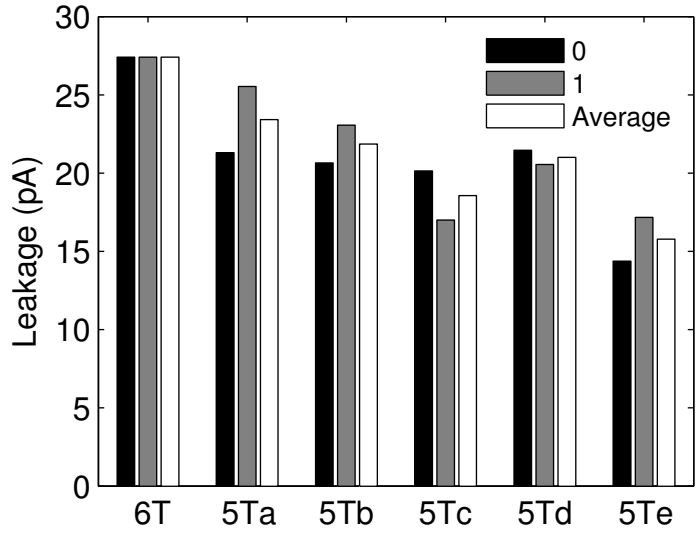


Figure 2.9: Leakage comparison between the 6T and the various 5T bitcells at 0.7 V.

degrading not more than 2% for the bitcells compared. Further, the worst case HSNM can actually improve with certain sizing approaches due to a reduction in the variability. For instance, we observe that the worst case HSNM improves by 7% for 5Tc and 4% for 5Td.

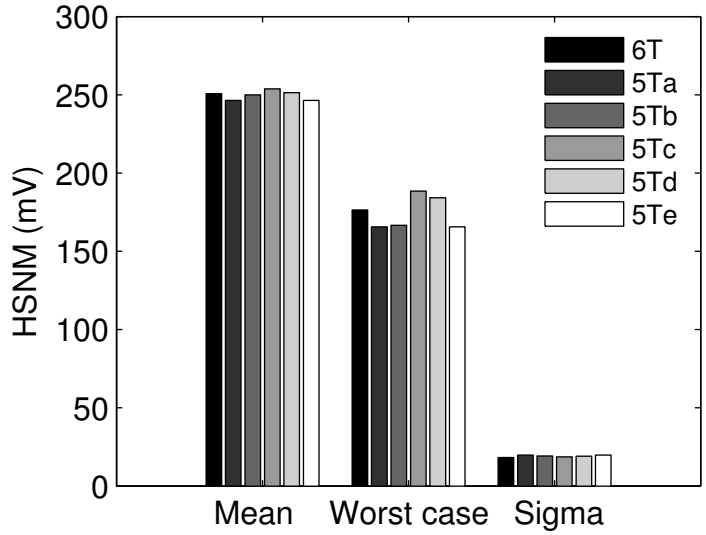


Figure 2.10: Comparison of the mean, standard deviation, and worst case HSNM of the 5T and 6T from a 1000 point MC at 0.7 V.

2.4.5 Write margin

The main drawback of the 5T bitcell is an inferior write margin due to the difficulty in writing a '1'. This can be rectified using write assists as seen in section 2.3.2. Fig. 2.11 shows the mean and sigma of the write margin for the 5T and 6T bitcells at 0.7 V. We measure the write margin by dc sweeping the WL voltage, with the BL held high, and the cell initially storing a zero. For all the 5T cells, we use a V_{DD} droop of 0.25 V, and a WL boosted by 0.2 V. The 6T cell is written without any assist. The difference between the boosted WL voltage and the cross point of the storage nodes is the write margin [37]. We observe that with sufficient write assist voltage bias, write margin can be recovered for all the 5T bitcells.

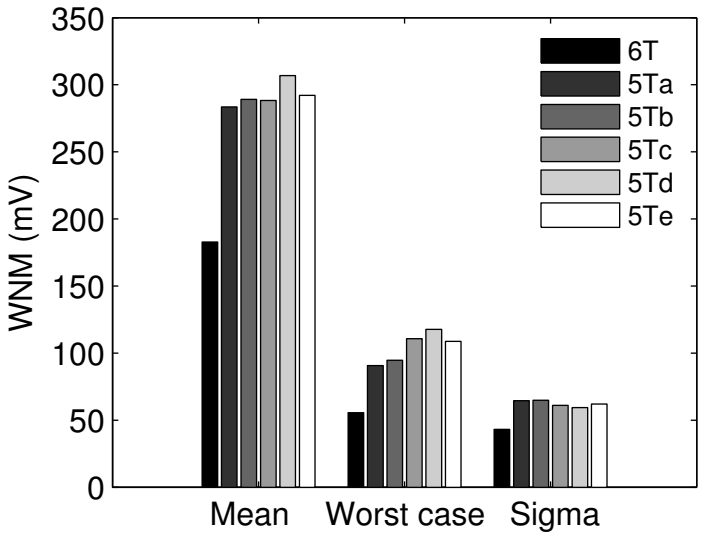


Figure 2.11: Comparison of the mean, standard deviation, and worst case WNM of the 5T and 6T from a 1000 point MC at 0.7 V.

The amount of write margin recovered through assist methods again varies with the asymmetric sizing approach used. For example, we see that 5Td has better mean and worst case write margin than the remaining 5T bitcells. This is because the cell is sized so that

N1 is not strengthened as much with respect to N3 as in the other cells, except 5Tc. At the same time, the trip point of P2-N2 is much lower for 5Td than 5Tc, meaning that it trips faster.

2.5 Comparison with 8T

The 8T bitcell (Fig. 2.12) adds a two transistor read stack to eliminate read disturb, which makes the 8T RSNM equal to the HSNM. Further, by sizing up the read stack and using short local bitlines, the 8T can provide high performance even though the sensing is single-ended. Due to its high stability and performance, especially at lower voltages, 8T SRAM has become a popular choice for lower-level caches in recent processor implementations [21][22][23].

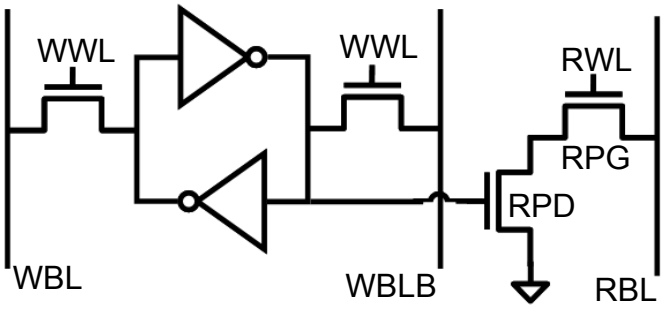


Figure 2.12: 8T bitcell schematic.

The drawback of the 8T is the area overhead, with the bitcell about 30% larger than the 6T. However, the macro area penalty is only about 7% for equal performance [16]. Further, the separate read port does not help eliminate read disturb for half-selected cells (e.g. WWL on, WBL and WBLB precharged high in Fig. 2.12) during a write operation. This is worked around either by not using column-multiplexing (e.g. [16]) or by using a

write-back scheme (e.g. [25]). The first solution makes the 8T SRAM susceptible to soft errors due to particle strikes, while the latter adds area, performance and power overhead.

In this section, we compare our proposed 5T to the 8T. For this comparison, we consider two 8T cells (Table 2.4). The first, $8T_{MIN}$ is composed of all minimum-sized devices. The second, $8T_{BIG}$ has a read stack that is sized up similar to the drive and access transistors for the reference 6T cell in Table 2.1.

Table 2.4: Normalized 8T sizing

	$8T_{MIN}$	**$8T_{BIG}$**
Pull-up (W_{PU}/L_{PU})	1.27/1	1.27/1
Pull-down (W_{PD}/L_{PD})	1.27/1	1.27/1
Pass-gate (W_{PG}/L_{PG})	1.27/1.04	1.27/1.04
Read Pull-down (W_{RPD}/L_{RPD})	1.27/1	4/1
Read Pass-gate (W_{RPG}/L_{RPG})	1.27/1.04	2.91/1.04

Fig. 2.13 shows the layout cartoon of the 8T bitcell. Using the same pushed rules as the reference 6T, described in section 2.2, the width of the 8T, X_{8T}, can be estimated from equation (2.4), where X_{6T} refers to the width of the 6T portion of the bitcell, as estimated using equation (2.1). The height of the cell remains the same. $8T_{MIN}$ has a width of 21.31, resulting in a 13% bitcell area overhead. $8T_{BIG}$ has a width of 24.04, resulting in a 27% overhead.

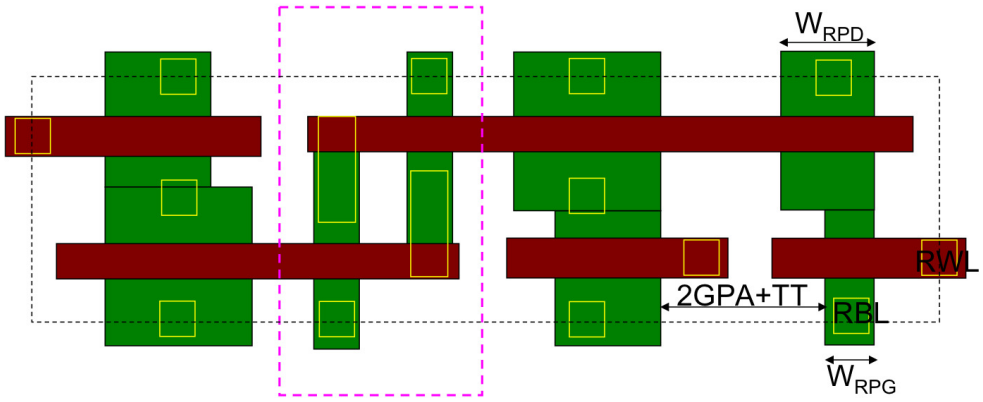


Figure 2.13: 8T bitcell layout cartoon.

$$X_{8T} = X_{6T} + max(AA, 2.GPA + TT) + max(W_{RPD}, W_{RPG}) \tag{2.4}$$

We start with the bitcell 5Ta and size up N1 so that it is the same area as the 8T bitcells. We also consider a sizing that results in an intermediate cell with an area overhead less than $8T_{MIN}$. Also, we can size up the access transistor with no area penalty to further improve I_{READ} (e.g. similar to 5Te), with a slight reduction of the RSNM gains as seen in section 2.4.1.

Fig. 2.14(a) shows the worst case I_{READ} out of 1000 Monte Carlo iterations for the iso-6T, iso-8T and intermediate 5T cells, relative to the worst case of the reference 6T bitcell. We observe that the proposed 5T can be sized to have a larger I_{READ} than either the 6T or 8T for the same area. For instance, if N3 is sized up as in the 5Te bitcell, the worst case I_{READ} for the 5T is 1.5× larger than that of $8T_{BIG}$. Since the 8T and 5T are both single-ended, this translates to a superior overall read performance for a given sensing scheme. Fig. 2.14(b) shows the worst case RSNM out of 1000 Monte Carlo iterations for the different bitcells. The RSNM of the 5T is lower than that of the 8T, whose RSNM is

the same as its HSNM, but higher than that of the 6T. For instance, the worst case RSNM of the 5T that has the same area as $8T_{BIG}$ is $2\times$ that of the 6T, but less than half that of $8T_{BIG}$.

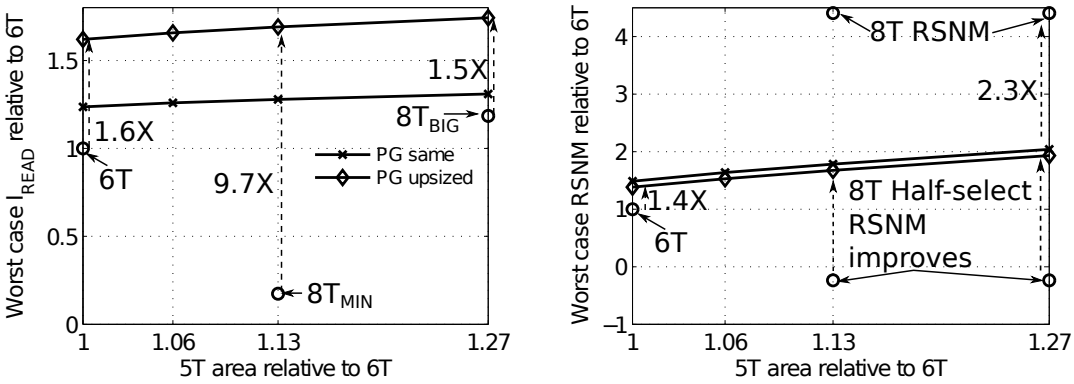


Figure 2.14: (a) I_{READ} and (b) RSNM comparison for iso-area 5T and 8T cells at 0.7 V from a 1000 MC simulation.

We observe that a 5T bitcell of intermediate area between a 6T and a 8T (6% larger than the 6T) can still have better I_{READ} than the smallest possible 8T cell (e.g. $8T_{MIN}$) and an RSNM 50% larger than the 6T. Further, during a write-operation, the RSNM of the half-selected 8T cells is worse than even that of the reference 6T since the pull-down transistors are sized down. Asymmetric sizing ensures that the 5T bitcell is much more robust to half-select read disturb than the 8T bitcell during a write operation. In summary, the 5T can be a high performance intermediate alternative between the 6T and 8T bitcells, especially in technologies where the SRAM V_{MIN} is read-limited.

2.6 45nm Test Chip

2.6.1 Description of Structures

A bulk CMOS test chip in a commercial 45nm technology was implemented. Fig. 2.15 shows the die photo. Two 16 kb 5T arrays , divided into 4 kb banks, each with a different asymmetric sizing scheme were fabricated along with a 16 kb conventional 6T array for reference. The main purpose of the test chip was to verify functionality of the 5T SRAM on silicon rather than targeting specific performance or power values.

The two 5T structures have bitcells with different asymmetric sizing approaches to exercise the SRAM metric trade-offs differently. The bitcell in bank 5T1 has N1 and P2 wider than N2 and P1 respectively, which strengthens inverter P1-N1 in Fig. 2.2. Bank 5T2 has N2 and P1 longer than N1 and P2 respectively, which weakens inverter P2-N2. Both banks have 128 cells per BL and the whole row is written or read at once (e.g. there is no column-muxing).

Fig. 2.16 shows the schematic of a 4 kb 5T block on the chip. A transmission gate multiplexer is used to switch between the normal and low cell voltage levels (V_{DDNOM} and V_{DDL} respectively). The WL voltage (V_{DDWL}) is also separate to allow various terminal voltages to be set independently. To simplify testing, the voltage supplies are all provided externally, rather than generated internally.

To read the single-ended 5T bitcell, a full-swing scheme with an inverter is used to 'sense' the BL, but other single ended sensing mechanisms, such as the pseudo-differential technique discussed in chapter 4, or the non-strobed regenerative SA [38] can be used to improve read speed.

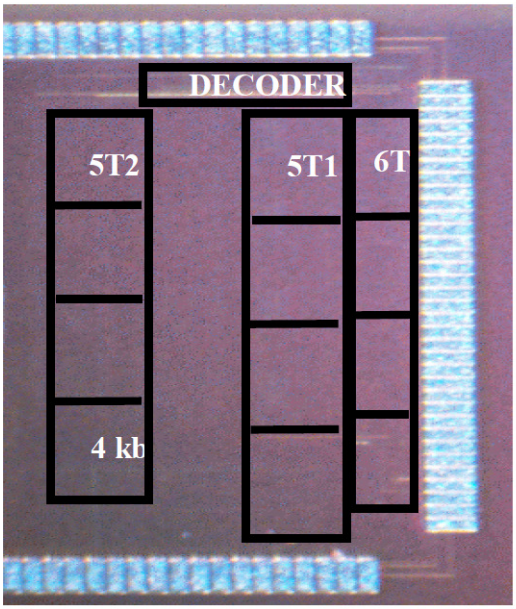


Figure 2.15: 45 nm bulk CMOS 5T test chip [29]

2.6.2 Measurements

To verify the improvement in the RSNM for the 5T over the 6T, we attempted to compare the read failures at lower voltages. However, both the 5T and 6T read correctly down to 0.5 V, where pad ring issues limit further measurement. The reason both bitcells were robust to such a low voltage is possibly due to the upsizing of the transistors to meet logic design rules since we were not permitted to push the rules, as is the norm. Nevertheless, this measurement confirms a robust read operation for the 5T at lower voltages.

The next focus of the measurements was to verify if sufficient writability can be recovered for the 5T through various circuit asssist techniques. Simply collapsing V_{DDC} can help recover the writability of the 5T. For instance, Fig. 2.17 shows that the 5T has full functionality at the nominal voltage of 1 V by using a drooped cell voltage during the write.

Next, we measured the impact of the sizing approach on the effectiveness of the write

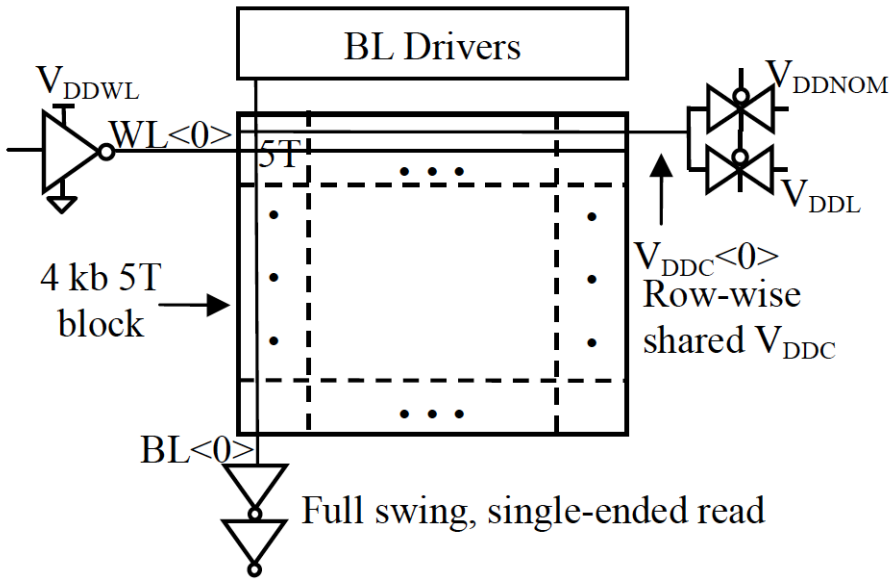
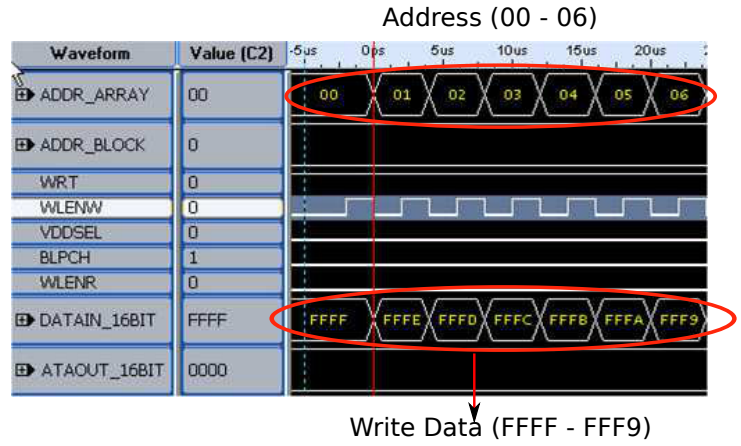


Figure 2.16: Schematic of 4 kb 5T block with write assist implementation [29].

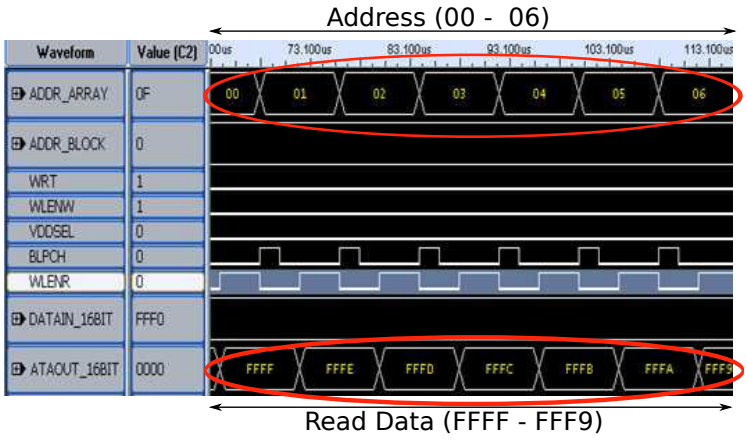
assist. We write all '0' and all '1' patterns to each row of a 4 kb array by collapsing cell voltage from 1 V to different values (V_{DDL}). We then read each row back at 1 V and determine the number of failed writes (Fig. 2.18). We observe that longer N2 and P1 (sizing 2) result in better write-ability (e.g. lower number of erroneous bits) than using wider N1 and P2 (sizing 1). This is because a wider N1 makes it harder for the access transistor to overcome the pull-down to write a '1'. This is an example of how a different asymmetric sizing approach can lead to a different trade-off in an SRAM metric, WNM in this case.

Using multiple write assists in conjunction further improves writability of the 5T. For instance boosting the WL voltage in addition to collapsing V_{DDC} during the write enables the cell to be written with a smaller drop in V_{DDC}. As Fig. 2.19 shows, when the WL is boosted to 1.2 V, a collapsed V_{DDC} value of 0.75 V is required to write the array with no errors. With no WL boost, the V_{DDC} needs to be dropped to 0.6 V to write with no errors.

Finally, we studied how well the write assists work at lower voltages. For this, we



(a)



(b)

Figure 2.17: Measured logic analyzer waveforms showing (a) write and (b) read operations at 1V.

measured the total bit error rate when writing the SRAM at lower voltages. Fig. 2.20 shows the error rate for the 5T and 6T arrays as voltage is scaled. With the help of WL boost of 20% of V_{DD} and V_{DDC} collapse to 50% of V_{DD}, the 5T demonstrates similar writability to the 6T up to 0.7 V.

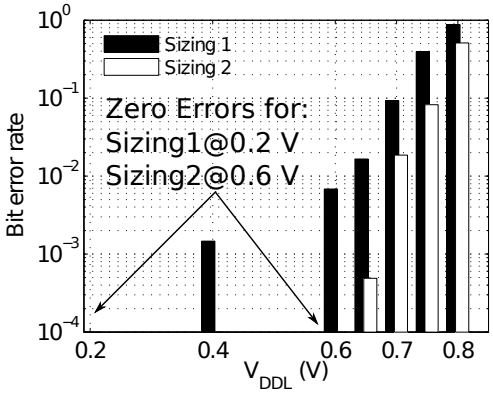


Figure 2.18: Impact of asymmetric sizing on write assist effectiveness. Measurements from a 4 kb array for each sizing are shown. Bit error rate is the number of observed bit fails divided by the size of the array.

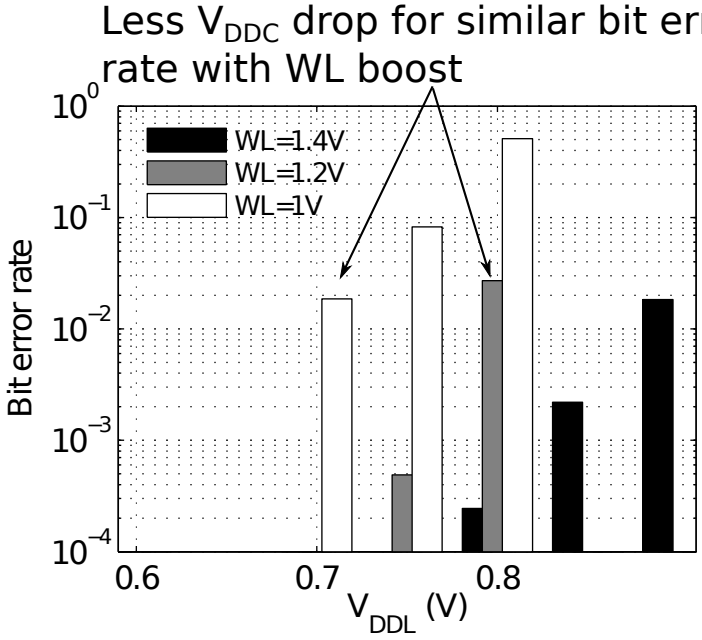


Figure 2.19: Measurements from a 4kb bank showing the use of multiple write assists to improve writability.

2.7 Conclusions

A 5T bitcell that provides improved read stability compared to 6T has been presented. By using asymmetric sizing as a knob, different metrics can be traded-off more effectively.

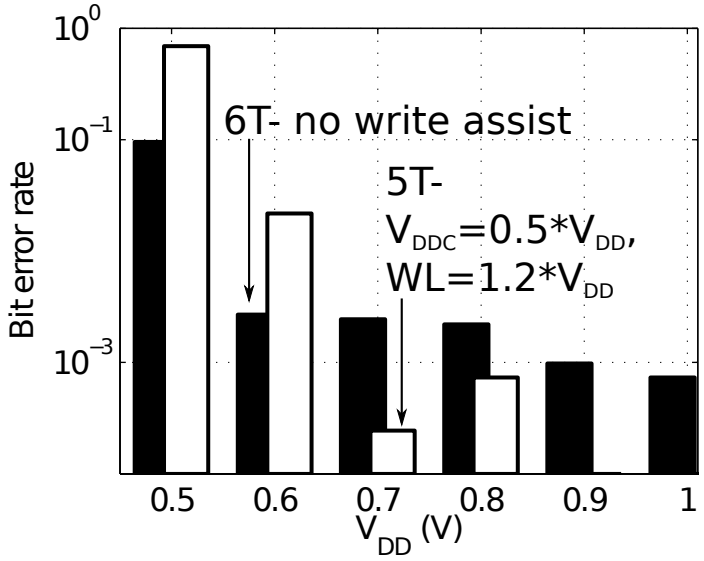


Figure 2.20: Measured bit errors for 4 kb 6T and 5T banks with write assist across voltage.

This enables the 5T to be a good intermediate candidate in the design space between the 6T and the 8T. Measurements from a test chip have demonstrated a functioning 5T SRAM. However, the main drawback of the 5T is its inferior WNM inspite of the write assists which make it unsuitable for near threshold voltages. Another drawback is the absence of a fast sensing scheme for single-ended bitcells that can exploit the potentially higher $\mathrm{I_{READ}}$ of the 5T bitcell. The asymmetric 6T bitcell and pseudo-differential sensing schemes presented in the next two chapters attempt to address these drawbacks.

Chapter 3

Asymmetric Six Transistor Bitcell

3.1 Motivation

The 5T bitcell described in the previous chapter provides flexibility in trading-off several SRAM metrics at the cost of degraded WNM. Though writability can be partially recovered through write assists, it still imposes a limitation on the 5T, especially in write-limited process technologies. In this chapter, we propose an asymmetric 6T bitcell that gives up a potential area advantage to recover and improve writability over the 6T using a two-phase reset and write operation.

3.2 Related Work

An asymmetric 6T with a single word-line (WL) was proposed in [39] that improves both RSNM and WNM at the cost of area. This cell is highly susceptible to half-select related failures. The authors overcome this problem by using a fine-grained bitline seg-

mentation scheme, which ensures that the disturb period is small. Thus the probability of an upset reduces. However, this scheme affects the area efficiency of the array due to the increased number of local write and read circuits. The asymmetric 6T that we propose has dual word lines that ensure that the RSNM for all disturbed cells during a read is improved relative to an iso-area symmetric 6T.

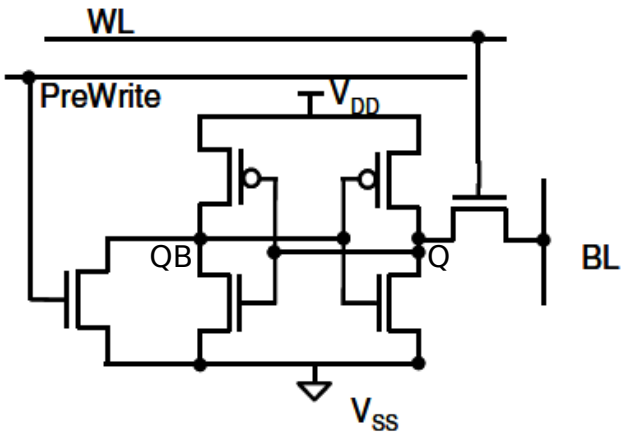


Figure 3.1: 5T with reset transistor [40].

The two-phase reset and write operation used in the proposed bitcell is similar to the method used in [40]. Fig. 3.1 shows their bitcell, which is essentially a 5T with a sixth transistor used to first reset the complementary storage node QB to '0', which writes a '1' to the cell. A single-ended write operation then follows to write '0's to the required cells. However, we can observe that with this implementation of the reset operation, all the cells in a row need to be reset at once. It also does not use asymmetric sizing, thus failing to capitalize on the RSNM benefits of asymmetric sizing, that we have explored in depth in the previous chapter. We modify their scheme so that the cell reset can be performed without having to reset the entire row, while simultaneously exploiting the benefits of asymmetric

sizing.

In the remainder of this chapter, we describe the proposed asymmetric 6T bitcell and compare it with the conventional 6T. In chapter 4, we explore single-ended sensing for the two bitcells proposed in this dissertation.

3.3 Dual WL Asymmetric 6T Bitcell

The asymmetric 6T bitcell has two key features. First, the writability is regained not through assist methods such as V_{DDC} droop or WL boosting as in the 5T, but using a two phase reset and write operation. Second, the same asymmetric sizing idea is exploited that was described for the 5T in the preceeding chapter. However, the reinstatement of the sixth transistor implies that the flexibility in trade-offs due to asymmetric sizing is reduced. This is because we want to keep the area of the bitcell the same as the reference 6T. The following sub-sections describe these two key ideas and the sizing of the bitcell that allows the exploitation of these ideas while keeping the same area as the conventional 6T reference cell.

3.3.1 Improving WNM and RSNM

Fig. 3.2 shows a generalized version of our proposed asymmetric 6T bitcell. The gate of the reset transistor, NR, is controlled by RSTG and the source by RSTS. RSTS is precharged high and RSTG is held low. As shown in the timing diagram in Fig. 3.3, before a write, RSTS pulses low and RSTG pulses high and thus resets the cell to a 1. RSTG and RSTS can be shared row-wise and column-wise, respectively, or vice-versa,

and activated only for those rows and columns in the array that contain the accessed cell (see Fig. 3.3). This eliminates the need to reset an entire row.

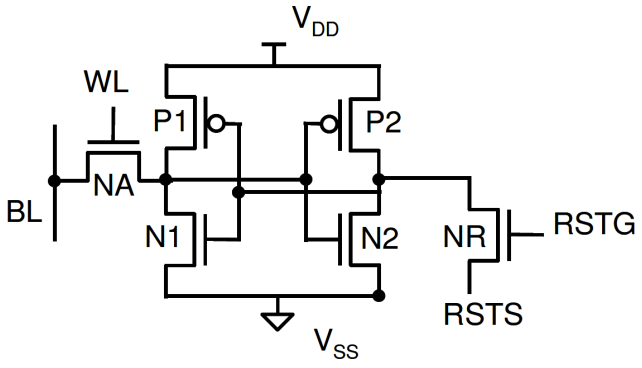


Figure 3.2: Proposed cell reset scheme.

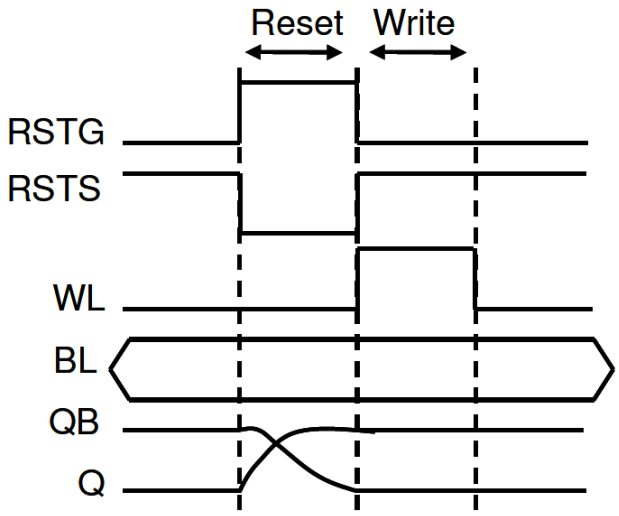


Figure 3.3: Timing diagram for cell reset and write.

If RSTG and RSTS are routed as shown in Fig. 3.4(a), the cells on the same column as the accessed cell are in a half-selected state during the reset operation. On the other hand, the cells on the same row as the accessed cell are in a half-selected state, if the reset signals

are routed as shown in Fig. 3.4(b). The routing method chosen does not affect the write functionality, and it still happens in two phases as described earlier. However, it has an impact on the area of the cell as we will discuss in the section 3.3.3.

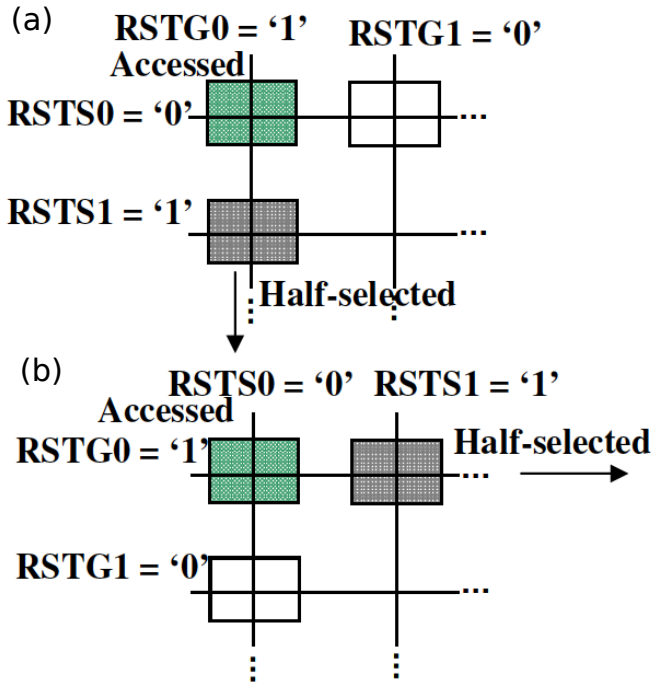


Figure 3.4: Options to share RSTG and RSTS for the reset transistor (a) column-wise and row-wise (b) row-wise and column-wise respectively.

Now that the writability is recovered through the reset operation without the need for an assist method, we can introduce asymmetric sizing in the cross-coupled inverters to improve RSNM and trade-off other SRAM metrics as for the 5T. Since it is possible to achieve a considerable improvement in RSNM (e.g. 42% for the 5Ta bitcell), we trade-off some of this improvement to instead improve the WNM. We do this by upsizing the access and reset transistors. In particular, upsizing the reset transistor is necessary to reduce the duration of disturbance to half-selected cells during the reset operation, since asymmetric sizing makes this end of the cell susceptible to upsets, as discussed in section 3.4.3.

3.3.2 Routing the Reset Signals

As described previously, there are two options to route the signals that control the reset transistor. The scheme in Fig. 3.4(a) increases the area of the cell even if we choose the same device sizes as the conventional 6T. This is because the RSTG contact that is analogous to a WL contact in the conventional 6T cannot be shared with an adjacent cell on the same row, as in the conventional 6T. Similarly the RSTS contact that is analogous to a BLB contact cannot be shared with an adjacent cell in the same column. This increases the bitcell width and height by the minimum distance that needs to be maintained between two polysilicon and diffusion regions (δw and δh respectively in Fig. 3.5).

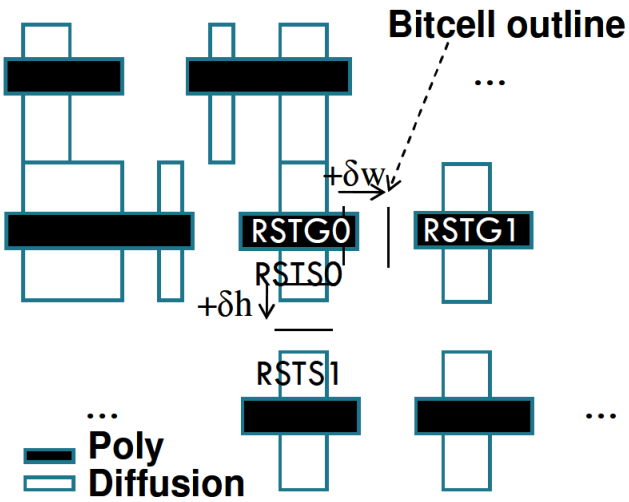


Figure 3.5: Layout cartoon for asymmetric 6T with RSTG and RSTS routed column-wise and row-wise respectively.

We selected the scheme in Fig. 3.4(b), which enables the RSTG and RSTS nodes to be shared similar to a WL and BL respectively, as in a conventional 6T. As the schematic in Fig. 3.6 shows, we rename the nodes RSTG and RSTS to WLR and BLR respectively, since they are routed similar to a WL and BL of a conventional 6T. In the remainder of this

chapter, this is the schematic we will refer to for the asymmetric 6T. In the next sub-section, we discuss how this bitcell is sized for the same area as the conventional 6T bitcell.

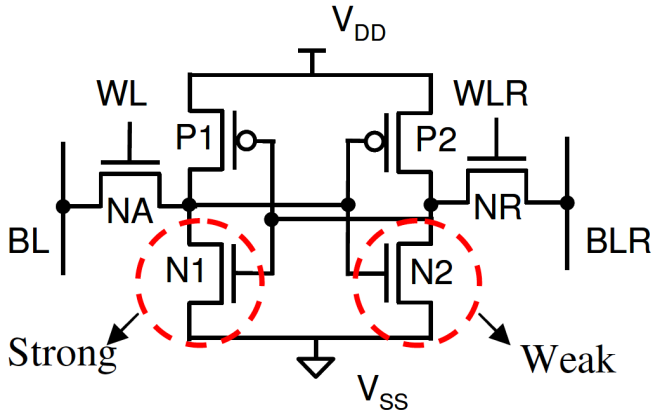


Figure 3.6: Asymmetric 6T schematic.

3.3.3 Sizing for Iso-area

Fig. 3.7 shows the layout of a conventional 6T bitcell and our asymmetric 6T. We did not change the lengths of any transistors since the commercial sub-45nm technology that we used did not allow it. Thus, the height of the two cells remains the same. For the conventional 6T, N1 and N2, P1 and P2, NA and NR in Fig. 3.6 are identically sized. We make the following changes to implement the asymmetric bitcell. First, we make the inverter P1-N1 n-strong by increasing W_{N1} relative to the reference 6T. We also size up the access transistor NA but keep it smaller than N1. Next, we reduce W_{N2} to the minimum width, which makes P2-N2 p-strong. This skewing of the cross-coupled inverters improves the RSNM. Finally, we increase W_{NR} so that the reset pulse can be of short duration. W_{NR} is increased so that the width of the cell doesnt exceed the width of the reference 6T, thus ensuring that the area of our cell is the same as the reference 6T.

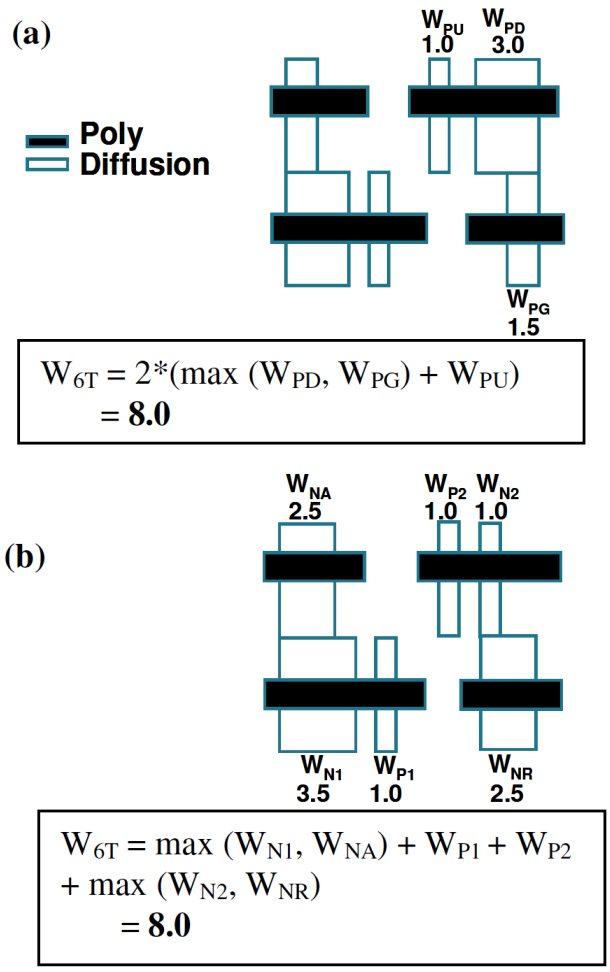


Figure 3.7: Layout cartoons with normalized device widths for (a) conventional (b) asymmetric 6T bitcells.

3.4 Comparison with 6T

In this section, we compare the asymmetric 6T with the conventional one in terms of RSNM, WNM, half-select stability, I_{READ}, and bitcell leakage.

3.4.1 Noise Margins and V_{MIN}

Fig. 3.8 shows simulation results in a commercial sub-45 nm technology for the mean RSNM and WNM of our bitcell and the conventional 6T. The margins are calculated using the butterfly curve method and the WL sweeping method, as was done for the 5T. We observe that the mean values of both read and write noise margins improve across different process corners and also at a lower voltage (e.g. 0.7 V). The mean RSNM improvement ranges from 10% (at SS, 1 V) to 22% (at FF/SS 0.7 V), while the WNM improvement ranges from 2% (at SS, 0.7 V) to 29% (FF, 1 V). Note that this improvement is obtained for the same bitcell area as the conventional 6T.

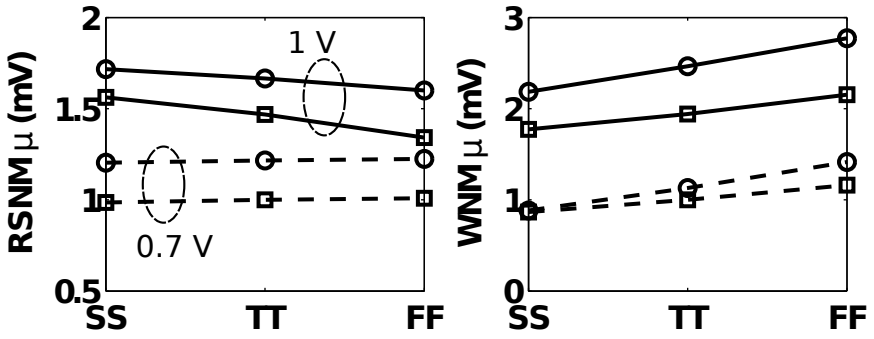


Figure 3.8: Normalized Mean (a) RSNM and (b) WNM. The circular and the square markers represent the asymmetric cell and the conventional 6T respectively.

Sizing up the access transistors has the additional benefit of reducing the variability of the WNM. As Fig. 3.9 shows, the σ/μ of the WNM is lower than that of the conventional 6T across different process corners and voltages, with a reduction of 36% at the FF corner, for the write '0 case at 1V. The variability of the RSNM increases, but the worst case RSNM for the asymmetric 6T is still better due to the same reasons discussed for the 5T in section 2.4.1.

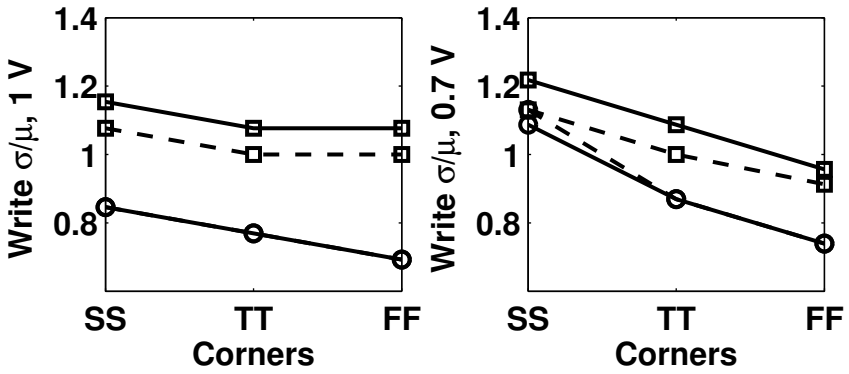


Figure 3.9: Normalized σ/μ of WNM at (a) 1 V and (b) 0.7 V at various process corners and voltages from a 1000 point MC simulation. The circular and the square markers represent the asymmetric cell and the conventional 6T respectively. The solid and dashed lines represent the write '0 and '1 cases respectively.

The larger noise margins result in a smaller V_{MIN} for our cell. We define the Read (Write) V_{MIN} for a bitcell as the minimum V_{DD} at which the cell can read (write) successfully. The Read (Write) V_{MIN} for an array is the maximum of the bitcell Read (Write) V_{MIN}. For a 1000 sample Monte Carlo simulation, we find the Read V_{MIN} for the asymmetric 6T to be 21.5% smaller and the Write V_{MIN} to be 4% smaller than the conventional 6T at the TT process corner.

3.4.2 I_{READ}

The wider pull-down and access transistors lead to an increase in read current. The mean read current increases roughly by 50% for the sizes chosen in our implementation. In addition, the variability of the read current in terms of σ/μ also reduces by 40% due to the larger access devices. This is a significant benefit when designing large memories, since the designer needs to look farther out in the tail of the distribution. However, this improvement in read current can be translated to a corresponding improvement in total read delay only

if a single-ended sensing scheme that has a comparable performance to differential sensing is used. We propose such a sensing scheme in chapter 4.

3.4.3 Half-select Stability

The stability of half-selected cells is reduced when compared to an unaccessed cell, both during read and write. An asymmetric 6T cell with similar sizing but a single WL has a worse half-select RSNM during both read and write. Fig. 3.10 shows the half-select state during read and write for both these bitcells. For both the asymmetric cells, due to the skewed sizes, the half-cell not involved in the read (e.g. P2-N2-NR in our case) is weaker than the half-cell of the conventional 6T, and is more susceptible to flipping. However, with the dual word-line scheme, NR is turned off during a read and the half-select RSNM is the RSNM of the stronger side (e.g. P1-N1-NA). Thus we reduce the half-select issue during a read when compared to a single-WL asymmetric cell as well as a conventional 6T.

As shown by the distributions in Fig. 3.11, the mean RSNM for half-selected cells during read improves by 18% under nominal operating conditions. The spread of the distribution increases, but the RSNM of the asymmetric 6T is always greater than that of the conventional 6T, as explained in section 2.4.1.

However, as Fig. 3.10 shows, the problem still remains during reset. We can get around the half-select problem by writing the entire row or by using a read-modify-write technique [13]. Thus an alternative implementation of our proposed bitcell would involve using a single WL, while keeping the same device sizes, and use one of the above approaches to solve the half-select issue. Another solution would be to make NR weaker by using a higher V_T device or by reducing its width. Since the half-cell P2-N2-NR is already sized

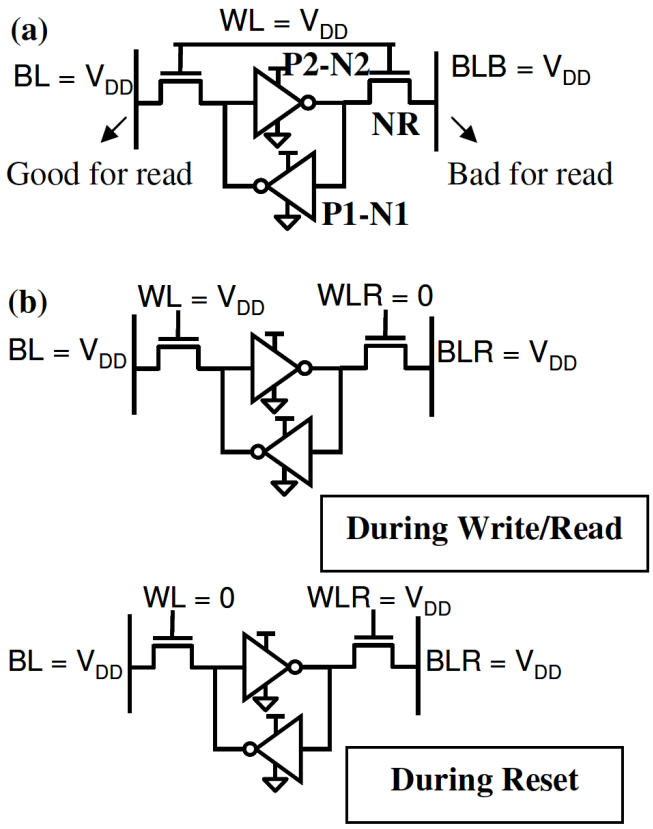


Figure 3.10: Half-selected cell for (a) single WL and (b) dual WL asymmetric 6T.

for write, we need not make NR as wide as NA. While this would also reduce the cell area, we chose not to do this since it would reduce the gains in write margin for the write '1' case and worsen the impact of variation.

3.4.4 Bitcell Leakage

Fig. 3.12 shows the leakage of the asymmetric 6T and the conventional 6T versus the supply voltage. Similar to the 5T, asymmetry causes the cell leakage to be data dependent. Compared to the leakage of the 5T (section 2.4.3), the asymmetric 6T leakage is higher due to an additional BL leakage component through the reset transistor. However, the average

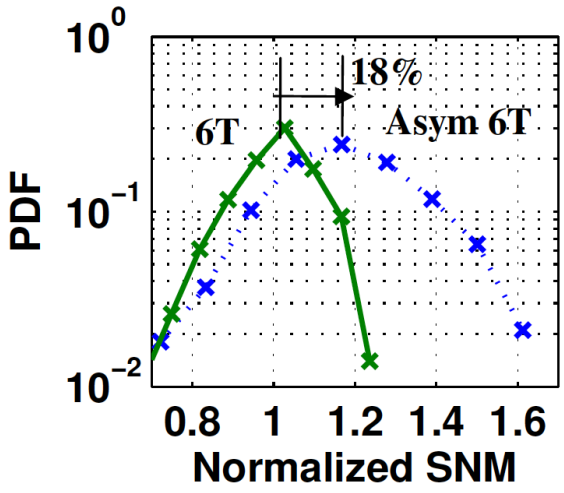


Figure 3.11: Half-select cell RSNM during read for conventional 6T, and asymmetric 6T at 1V, TT, and 27°C from a 1000 point MC simulation.

leakage of the asymmetric cell is nearly the same as that of the conventional 6T at any particular voltage. Moreover, the increased robustness of the asymmetric 6T cell enables a lower V_{MIN}, which implies that the asymmetric 6T array would have a lower standby leakage than a conventional 6T array when operating under V_{MIN} conditions. We further observe that the leakage of a cell storing '0' is less than that of the conventional 6T by 10-15%, due to the smaller size of the leaking pull-down (N2). Since SRAM caches tend to store more number of 0s than 1s [41], the leakage power of the asymmetric 6T array at a given operating voltage could potentially be less than that of a conventional 6T.

3.5 Comparison with 8T

In chapter 2, we saw how the 5T bitcell can be an intermediate point in the design space between the conventional 6T and the 8T in terms of performance (e.g. I_{READ}) and stability (e.g. RSNM). Since the asymmetric 6T is based on the same concepts of RSNM improve-

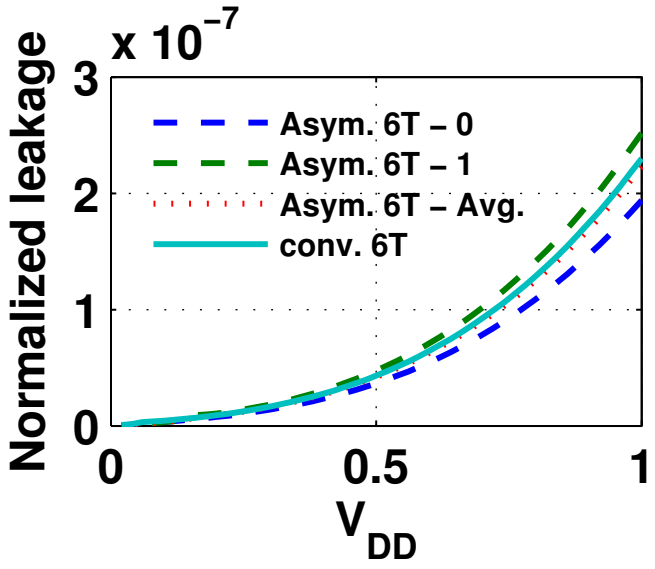


Figure 3.12: Standby cell leakage for conventional and asymmetric 6T.

ment and flexible trade-offs using asymmetric sizing, we can expect the asymmetric 6T to play a similar role as the 5T in the SRAM design space.

However, restoring the sixth transistor means that the extent of asymmetric sizing is reduced compared to the 5T if the same area is to be maintained as the conventional 6T or the 8T. On the other hand, the WNM is now improved compared to the conventional 6T and the 5T. Thus, compared to the 5T, the asymmetric 6T is a more balanced intermediate design point between the conventional 6T and the 8T. In other words, for the same bitcell area, the stability (e.g. both RSNM *and* WNM) and the performance (e.g. I_{READ}) can all be improved when compared to the 6T, but the RSNM and the I_{READ} can be improved to a lesser extent when compared to the 5T.

3.6 Test Chip

A test chip in a sub-45nm CMOS technology was fabricated to test the asymmetric 6T SRAM and the pseudo-differential sensing scheme. The chip comprised three SRAM macros – a 128x64 asymmetric 6T, a 64x64 conventional 6T, and a 128x64 conventional 6T, all with iso-area logic rule compliant bitcells. The asymmetric 6T macros uses the pseudo-differntial sensing scheme and also has chicken switches to use a fixed external voltage reference instead. The conventional 6T SRAMs use gate-input differential voltage SAs. While the 64x64 6T has the same number of bitcells per discharging BL as the asymmetric 6T (e.g. either the top or bottom half), the 128x64 has the same total number of cells per BL. The test chip has been available for wafer testing since Spring 2011. However, due to logistical issues it has not been tested yet.

3.7 Conclusions

We have presented an asymmetric 6T bitcell that has higher RSNM and WNM than the conventional 6T for the same bitcell area, and reduced variability in WNM. In addition, it provides higher I_{READ} with lower variability. This comes at the cost of reduced trade-off flexibility compared to the 5T, due to the restoration of the sixth transistor. However, improvement in RSNM, WNM and I_{READ} means that the asymmetric 6T is a more balanced intermediate bitcell between the conventional 6T and the 8T in the SRAM design space, when compared to the 5T.

Finally, improving only the I_{READ} may not sufficient to improve read performance, since it also depends on the sensing scheme used. In the next chapter, we propose a sens-

ing scheme that provides comparable overall read performance as conventional differential sensing.

Chapter 4

Pseudo-differential Sensing for Single-ended Bitcells

4.1 Motivation and Related Work

The alternative bitcells proposed in the previous two chapters require a fast single-ended sensing scheme to leverage the improvement in I_{READ}. Conventional single-ended sensing employs a hierarchical bit line structure with full-swing sensing as seen in recently proposed SRAMs in processor caches [21][22][23]. By using short local BLs and multiple sensing circuits periodically inserted in the array, this method trades-off macro area (e.g. 20-30% in [16], 8.5% in [25]) for improved performance and lower power. If the designer is willing to sacrifice macro area, hierarchical full-swing sensing is the clear choice for the 5T as well. Higher I_{READ} can also enable larger number of bitcells per column, thus

reducing the macro area penalty imposed by a hierarchical BL structure.

However, if sacrificing some area efficiency is not an option, an alternative single-ended sensing scheme is required. Thus, we propose a pseudo-differential sensing scheme for single-ended bitcells like the 5T and the asymmetric 6T. Similar schemes were proposed for single-ended register files [42] and for DRAM cells [43]. In this chapter, we describe our single-ended sensing scheme and analyze its pros and cons.

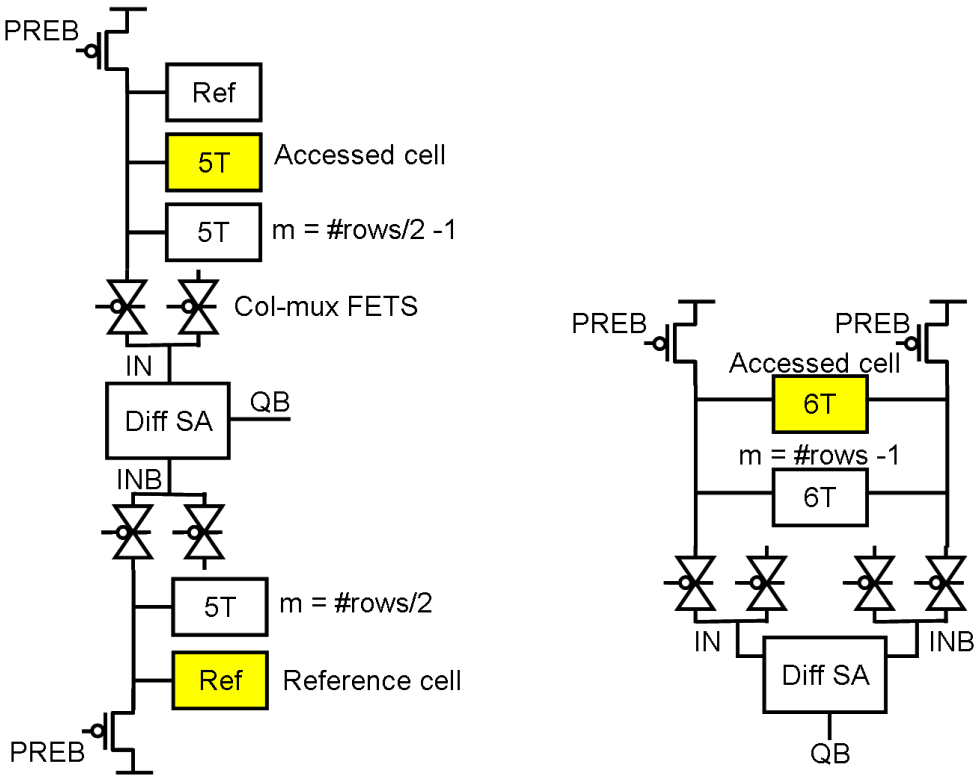
4.2 Pseudo-differential sensing

4.2.1 Overview

Fig 4.1(a) shows our sensing scheme. A differential SA is placed in the middle of the array, with each input of the SA connected to one half of the array. Weaker reference cells on either side help perform a pseudo-differential sensing.

The sensing works as follows. Without loss of generality, we assume that the cell being read is in the top half of the array. The WL of the accessed row is activated, along with the WL of the reference row on the opposite side of the SA. If the cell being read stores a zero, it discharges the corresponding BL faster than the reference cell, as shown by the curve marked '0' access in Fig. 4.2. On the other hand, if the cell being read stores a high value, the BL does not discharge, as indicated by the curve marked '1' access in the figure. The reference cell on the other side discharges as before. Thus, an appropriate differential is developed, which is quickly resolved by the SA. Note that the output of the SA would need to be inverted if the accessed cell was in the other half of the array.

The reference bitcells in this scheme have lower I_{READ} than the actual bitcells. This



(a) (b)

Figure 4.1: Column schematics for (a) 5T with pseudo-differential sensing and (b) 6T with differential sensing.

can be accomplished by designing a separate weaker cell, or by lowering the WL voltage of the reference row [32]. We use the latter option in this work. Since only two reference rows are required for the whole array, we consider the macro area overhead imposed by them to be negligible.

The authors in [26] also use a pseudo-differential sensing scheme for their 5T SRAM. However, it differs from our scheme in that they do not require an explicit reference cell since the BLs are not precharged to the full rail value. Consequently, the BL connected to the accessed cell either charges or discharges relative to this reference BL precharge voltage, thus providing the necessary differential for the SA. In our scheme, a reference

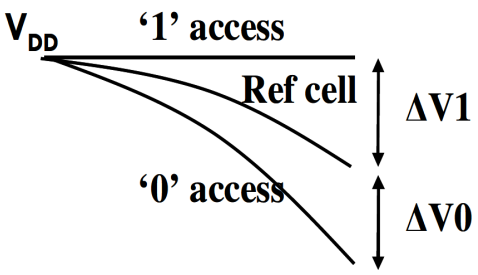


Figure 4.2: Bitline discharge cartoon for pseudo-differential sensing. ΔV1 and ΔV0 represent the BL voltage differential developed for reading '1' and '0' respectively.

cell with lower I_{READ} is needed to sense a '1'.

4.2.2 Pros and Cons

For the same total number of bitcells in the column, such a scheme roughly halves the BL capacitance compared to conventional 6T differential sensing. Combined with higher I_{READ} of the 5T cell due to asymmetric sizing, this scheme can compensate for the additional BL discharge that a pseudo-differential sensing scheme requires.

The main drawback of the pseudo-differential scheme is the increased susceptibility to variation due to the reference cell (see section 4.3). Also, with the data input and output now located in the center of the array, driving data in and out becomes a challenge. This can be overcome either by using higher layer metals to route the final data, or by routing the data laterally, with an area penalty that increases with word size.

In general, single-ended sensing is slower than differential sensing and more susceptible to noise on the BL due to the absence of the common mode noise rejection. However, this is beginning to change due to the increasing impact of BL leakage. In [36], the authors compare single-ended and differential sensing for symmetric and asymmetric cells under different leakage scenarios. They demonstrate that though single-ended sensing is slower,

high BL leakage and using asymmetric cells shrinks this gap. Another reason that can contribute to speeding up single-ended sensing is the absence of an external SA strobe and the variability associated with it. Thus, with technology scaling and increasing leakage and variation impact, it is conceivable that single-ended sensing can catch up with differential sensing.

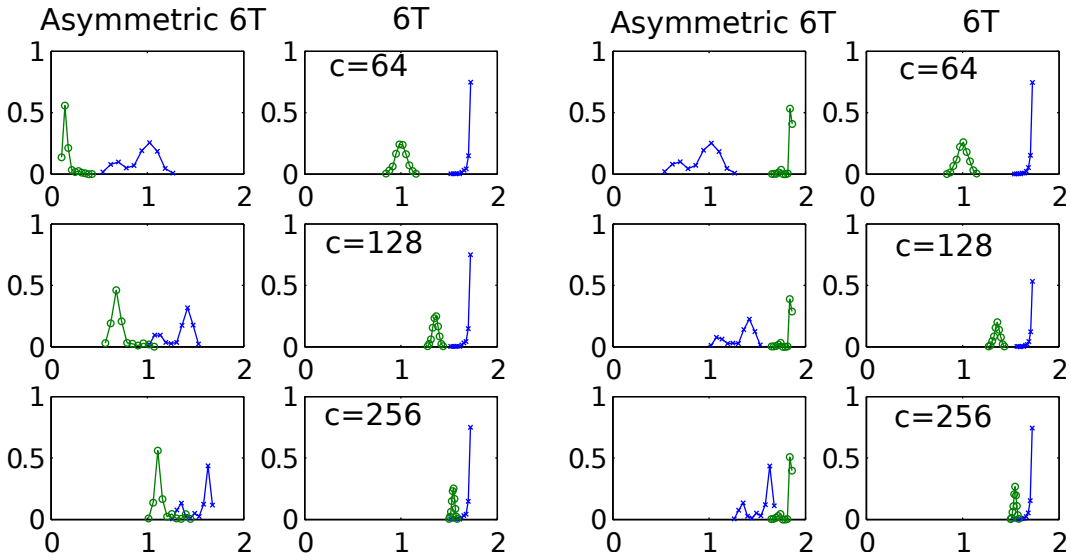
A non-strobed regenerative SA was proposed in [38]. The absence of a strobe and implicit offset cancellation enables this SA to have a better worst case read access time than the differential SA and this is could conceivably be used for reading the 5T. In this work, we restrict our comparison to the pseudo-differential scheme as having the same SA in either case makes a fair comparison of read speed easier.

4.3 Impact of Variation

In this section we look at the impact of variation on the pseudo-differential sensing scheme, using the example of the asymmetric 6T. The impact is expected to be similar for the 5T as well.

Since the reference voltage depends on the I_{READ} of the reference cell, variation in this cell, as well as bitline leakage, can cause the distributions of the SA input voltages to overlap. This leads a '0' and '1' to be indistinguishable. In conventional sensing on the other hand, the variability of the reference voltage is only due to bitline leakage. Fig. 4.3 shows the distributions of the SA inputs (IN and INB in Fig. 4.1), for a fixed WL pulse, and fixed reference voltage for pseudo-differential scheme. The distributions are shown for three different column heights (c = 64, 128 and 256) and for both read '0' and '1' cases. The column height here refers to the number of cells on the discharging BL. Thus, in the

split BL case, the total height of the array is 2c, while it is c for the differential sensing case.



(a) (b)

Figure 4.3: SA input distributions for (a) Read '0' and (b) Read '1'. The distributions with the circular and cross markers are associated with the sensed and reference BL voltages respecitvely.

We observe that as the height of the column increases, the overlap between the sensed and reference voltage increases, causing the SA output to be unreliable. However, for shorter column heights, as is the norm in modern SRAM designs, the distributions of the sensed and reference voltage are wide apart. This is because the higher drive strength and lower BL capacitance pulls the two distributions away from each other. Thus, the split BL sensing works for shorter BLs though the reference has higher variability when compared to conventional sensing. Note that the BLs are still longer than that used in full-swing hierarchical sensing (e.g. 8 BLs per local BL in [16]). Thus, the area efficiency is not impacted.

4.4 Comparison with Differential Sensing

Fig. 4.1(b) depicts the schematics of a column for differential sensing. Both the sensing schemes being compared have the same total number of bitcells per SA. The NMOS input devices of the SA are sized up since increasing their W and L reduces the offset variation [44]. We use the 5Te bitcell described in chapter 2 for our comparison. Note that this bitcell is tuned for maximum I_{READ} by upsizing both access and pull-down transistors. The read performance is expected to change depending on the asymmetric sizing scheme used.

4.4.1 Read Delay

Since the rest of the read path is the same, we can compare the read access time simply by comparing the time it takes to develop a sufficient BL differential. The sized-up SA that we use has a 3σ offset of 80 mV. For this example, we add another 20 mV to account for strobe uncertainty and measure the time it takes to develop 100 mV of differential in either case. Fig. 4.4 shows the mean and worst case delay for 5Te and 6T cells. Though the mean is comparable, the higher variation for the pseudo differential case makes it slower in the worst case then 6T. However, for larger arrays (e.g. higher number of rows), the 5T is only 15-20% slower than 6T. Further, if we use full-swing read for the 6T as well, then the 5T will be faster due to higher I_{READ}.

4.4.2 Read Power

Typically, the power during a write cycle is more than that during a read cycle due to the full-swing nature of the write operation. However, since reads are more frequent than

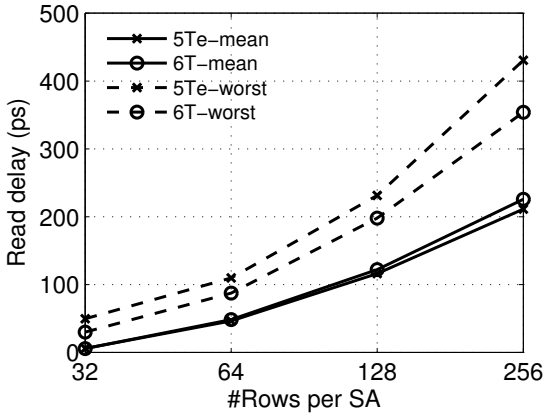


Figure 4.4: Read delay comparison for 5Te with pseudo-differential sensing and 6T with differential sensing at 1V.

writes, the read power becomes important. So, we analyze the read power per column for the 5T. The power here refers to that dissipated by the precharge and column mux transistors, the bitcell column, the BLs. Fig. 4.5 depicts the power per column for a read access. For the 5T, the average of the power when reading a '0' and '1' is shown. When reading a '0', both the top and bottom BLs discharge. Also, a higher droop is required for pseudo-differential read when the cell stores '0'. This contributes to increasing the read power relative to the 6T. However, this is offset by the halving of the BL capacitance. The net result is that the power is nearly the same (within 10%) for both cases, as seen in Fig. 4.5.

4.5 Conclusions

A pseudo-differential scheme for single-ended bitcells such as the 5T and the asymmetric 6T described in the previous two chapters has been presented. The proposed technique allows the improved I_{READ} provided by these asymmetric bitcells to be translated to better overall read delay, comparable to differential sensing. On the flip side, the scheme is more

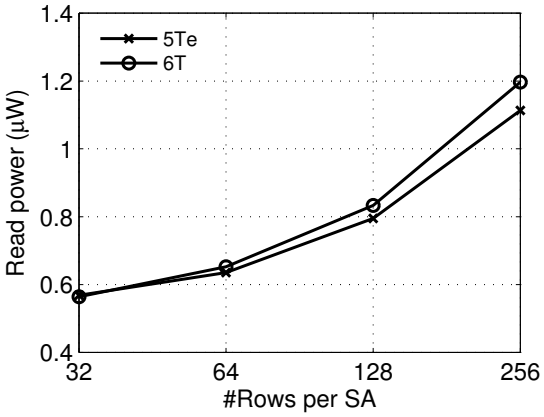


Figure 4.5: Read power per column for 5Te and 6T at 1V is similar.

susceptible to variation and provides comparable performance only for shorter BLs (e.g. 64-128).

Chapter 5

Dynamic Write-limited Minimum Operating Voltage for SRAM

5.1 Motivation and Related Work

An accurate prediction of the minimum SRAM operating voltage, V_{MIN}, is necessary for designing a low power SRAM that meets retention, read, and write functional yield requirements. Existing attempts at V_{MIN} prediction for standby and active operation, and for yield estimation are based on static margins during hold, read, and write operations [45][46]. However, static metrics tend to be optimistic for write and pessimistic for read, since by definition, the cell disturbance (e.g. WL pulse) is considered to be of infinite duration for evaluating these metrics. Thus, to be able to estimate V_{MIN} more accurately, it is imperative to consider dynamic margins (DMs) for cell read and write stability [7].

There have been several works recently that investigate SRAM read and write operations from a dynamic perspective. The authors in [7] investigate dynamic read and write

stability in terms of the separatrix, which divides the SRAM state space into two stability regions. The read and write DMs are defined as the margin between the T_{WL}, the width of the WL pulse, and T_{ACROSS}, the time taken to cross the separatrix. Dynamic read stability is investigated in [9] and takes into account the impact of repeated read accesses on the dynamic stability of the bitcell. In this dissertation, we focus on dynamic writability analysis alone since static write margin is optimistic and tends to underestimate V_{MIN}. Also, the PMOS pull-ups that influence the write operation are typically the smallest devices in the cell, and hence more impacted by variability. This makes write failure more likely, especially in newer technologies [47].

Although dynamic stability has been researched extensively in recent times, most of the work has been focused on defining DMs [7][48], devising ways to calculate them analytically [6] or on-chip [49], or calculating failure probability based on dynamic stability [8]. In this dissertation, we focus on how dynamic stability affects V_{MIN}. To achieve this goal, we first define dynamic write noise margin (DNM) based on the $T_{WL-CRIT}$ metric for dynamic writability proposed in [48] (called T_{CRIT} in [48]. We then relate this DNM to the dynamic write-limited V_{MIN} (DWV_{MIN}) and compare it with conventional Static V_{MIN}. To the best of our knowledge, this is the first such definiton of a V_{MIN} based on dynamic writability.

We observe that it is not possible to evaluate which measure of dynamic writability of the several proposed in literature and this work is the most accurate and effective in predicting the correct V_{MIN}. It is quite possible that our metric may not be the most ideally suited in certain scenarios. Irrespective of the actual dynamic metric used, the significant contribution of this work is that it accounts for the factors that cause the actual (e.g. dynamic)

writability of the SRAM bitcell to be different from that predicted using static metrics.

To enable lower power through lower V_{MIN}, writability can be improved by voltage-bias based assist techniques such as [50][51][52]. The impact of different assist techniques on write SNM [15] and on dynamic writability [53] has been investigated earlier. In particular, [53] investigates the efficacy of various assist techniques in reducing $T_{WL-CRIT}$. Our second contribution in this dissertation is to investigate the impact of write assist methods on the DWV_{MIN}.

5.2 Critical WL Pulse-width

In order to evaluate the writability of a cell more precisely, a metric which takes into account the dynamic write behavior must be used. We use the minimum WL pulse width ($T_{WL-CRIT}$) for the cell to flip ultimately to the correct new state as a metric for dynamic writability. The reason this is a suitable metric for dynamic writability can be expalained using the concept of the state space of the SRAM bitcell and the separatrix [54][6].

Fig. 5.1 depicts the state space of the SRAM bitcell, where Q and QB refer to the true and complementary states of the data stored in the cell. The cell has two stable states (0,1) and (1,0). The black curve in the state space represents the separatrix, which separates it into two regions of attraction. When noise injected into the cell disturbs the cell from one of its stable states, it is "attracted" back to the original stable state, provided the disturbance does not move the cell state across the separatrix. In that case, the cell is attracted to the other stable state, resulting in the cell flipping its stored data.

A write operation can be treated as a special case of noise injection in the cell. This noise can be treated as a current injection through the pass gates [48] during write. Thus,

for a successful write, the noise injected through the pass gates must be sufficient for the transient state trajectory to cross the separatrix. Note that the amount of noise injected depends both on the amplitude *and* duration of the WL pulse. As Fig. 5.1 depicts, when the duration is equal to $T_{WL-CRIT}$, the cell just flips (e.g. the red curve). If the WL duration is insufficient, the state trajectory is attracted back to the original state, implying that the write fails (e.g. the green curve).

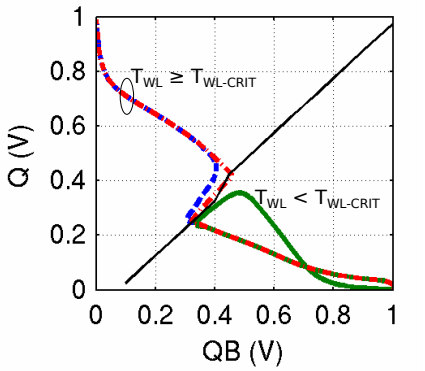


Figure 5.1: Transient state trajectories and the separatrix in the state-space demonstrating the relation between the $T_{WL-CRIT}$ and the dynamic writability of the bitcell. For $T_{WL} \geq T_{WL-CRIT}$, the state changes and the write is successful.

5.3 DNM and DWV_{MIN}

The $T_{WL-CRIT}$ described in the previous section is a measure of the dynamic writability of the bitcell. However, it is not a true "margin" and does not reflect how close to failure the cell is. We propose a new definition of dynamic write margin as $T_{WL} - T_{WL-CRIT}$, the difference between the WL pulse width and the critical pulse width required for the cell to flip. The larger the T_{WL} compared to $T_{WL-CRIT}$, the larger the margin, meaning that the cell is less susceptible to dynamic write failure.

We now define the DWV_{MIN} of a bitcell as the supply voltage at which its dynamic write

margin is less than a certain threshold (we use zero). It determines the extent to which V_{DD} can be lowered before the bitcell becomes dynamic write limited. The DWV_{MIN} of an array is determined by the bitcell that has the maximum $T_{WL-CRIT}$ and minimum T_{WL}.

We make two assumptions in our analysis of DWV_{MIN}. First, we assume that the variation of T_{WL} across the array is negligible when compared to that of $T_{WL-CRIT}$. This is justified because the WL pulse is driven by inverters that are usually made up of fairly large devices. Moreover, there are far fewer number of WL drivers than bitcells, which means that the spread in T_{WL} encountered on a chip will be much lower than that of $T_{WL-CRIT}$. Thus, we assume a constant T_{WL} for a given voltage. Second, we note that the the $T_{WL-CRIT}$ for a write '0' would be different from that for a write '1' due to local mismatch. Thus, the DWV_{MIN} is actually the maximum of the DWV_{MIN} of the write '0' and write '1' cases. In this work, we only look at the DWV_{MIN} corresponding to the write '1' case. Since the write mechanism is the same for both, we expect the same analysis to apply for the write '0' case as well.

If a cell is statically limited (e.g. static write margin is zero), the cell cannot be written even if T_{WL} is infinite. This happens when the variability within the bitcell is such that the pass-gate is severely weakened when compared to the pull-up on the side storing a '1'. We consider $T_{WL-CRIT}$ to be undefined for such statically limited cells as there does not exist a value of T_{WL} that would allow the cell to flip. Fig. 5.2 shows an example using an early bitcell from a 32nm low power, CMOS bulk technology. The pass gate threshold voltage (V_T) is 88 mV higher than nominal and the magnitude of the pull up V_T is 120 mV lower than the nominal value. As a result, the storage nodes (Q and QB) cannot be flipped when V_{DD} is below 0.686 V. This cell has a negative write SNM below 0.686 V, as measured

using Seevinck's method [30]. Such cells determine the static write limited V_{MIN} of the array.

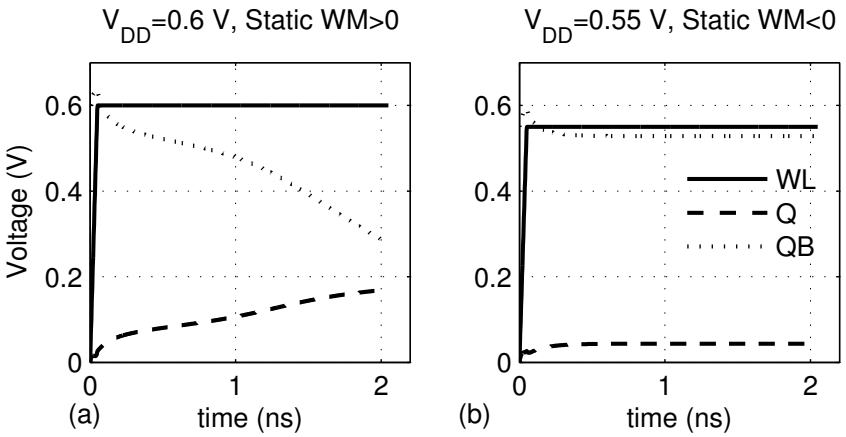


Figure 5.2: A dynamically write limited but statically non-limited cell (a) becomes statically limited (b) as the voltage is lowered from 0.686V to 0.55V.

Fig. 5.3 depicts the T_{WL} (for a given WL driver) and the worst case $T_{WL-CRIT}$ of a 1kb array. As the voltage is lowered, both T_{WL} and worst case $T_{WL-CRIT}$ increase. The latter does so more rapidly until the dynamic write margin becomes zero at the intersection of the two curves, 624 mV, which is the DWV_{MIN}.

The first static write failure appears at 670 mV (Fig. 5.3). If the array hits the static limit before it becomes dynamically limited as in this case, the DWV_{MIN} is irrelevant. However, to understand and characterize the dynamic writability phenomenon, we continue looking at the $T_{WL-CRIT}$ and DWV_{MIN} even after the array is statically limited.

We observe that there is a kink in the $T_{WL-CRIT}$ curve once the array becomes static write limited. This is because as V_{DD} is lowered, the weakest cell in a dynamic writability sense (e.g. with the largest $T_{WL-CRIT}$) starts becoming static limited instead (e.g. $T_{WL-CRIT}$ is not defined). This is confirmed by the fact that the V_T offsets are the same

for each voltage (Table 5.1). As a result, the worst case $T_{WL-CRIT}$ now corresponds to a relatively stronger cell than before (e.g. not as far out the tail), causing the kink in the curve. We also note that the pull-up and access transistor on the side storing a '1' are the worst affected by variability, being significantly strengthened and weakened respectively. Thus, the same devices influence both static and dynamic writability.

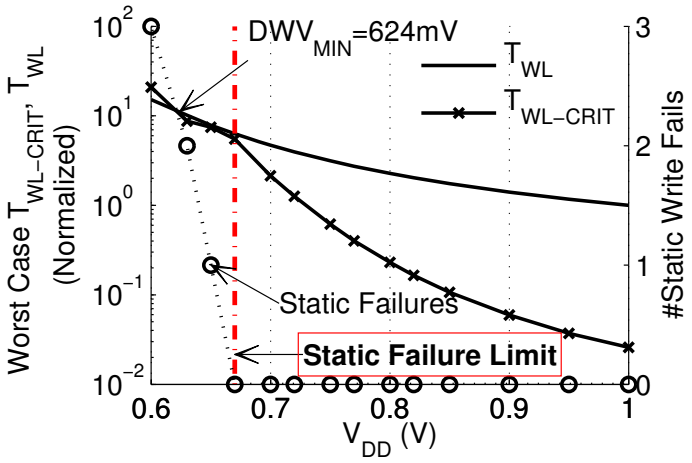


Figure 5.3: Using worst case $T_{WL-CRIT}$ and T_{WL} to determine the dynamic writability limited V_{MIN}. The intersection determines DWV_{MIN}, 624 mV in this case.

Table 5.1: V_T offsets for static and dynamic write fails

	PD0	PD1	PU0	PU1	PG0	PG1
1 V	0.0286	0.0332	-0.0245	-0.1294	-0.0258	0.1263
0.6865 V	0.0286	0.0332	-0.0245	-0.1294	-0.0258	0.1263

5.4 Factors affecting DWV_{MIN}

DWV_{MIN} for an array depends on four factors - the nature of the generated WL pulse (e.g. how it scales with voltage), the memory capacity, the number of cycles prior to first read, and the bitcell parasitic capacitances. We now discuss these aspects in detail.

5.4.1 WL pulse characteristics

Typically, the final WL pulse that is driven to the bitcell is generated by combining an enable pulse with the address decoder output to activate one row. This enable signal, along with other control signals such as the sense amplifier enable and precharge signals are generated by a timing block, for instance, using a self-timed replica path to track process variations or simply through combinational logic that depends on a clock input [55].

The DWV_{MIN} depends on how the generated WL pulse scales with voltage. Fig. 5.4 shows how the DWV_{MIN} changes for two different T_{WL} scaling approaches. In Fig. 5.4a, the WL pulse is generated using a self-timed path that traverses the height of the array. The T_{WL} values are derived from simulations of an extracted model of a heavily margined, compiler generated array. Fig. 5.4b, the T_{WL} at each voltage is set to the value that is required to ensure that a specific bitline differential is developed by the end of the pulse during a read. For this example, we arbitrarily choose a differential of 150 mV. This approach results in a much smaller T_{WL} across voltage when compared to the former approach. As a result, the array becomes dynamic write limited at a higher voltage. We observe that the DWV_{MIN} for a 1kb array is 624 mV with the former approach, while it increases to 741 mV with the latter.

In general, there are several factors that determine the T_{WL}. For example, it has to be

wide enough to generate a sufficient bitline differential to overcome bitline leakage and sense amplifier offset during a read. On the other hand, it has to be narrow enough to meet performance requirements and to avoid read upsets. Scaling T_{WL} so that it is larger than the worst case $T_{WL-CRIT}$ of the array will ensure that the array is not dynamically write limited at a particular voltage.

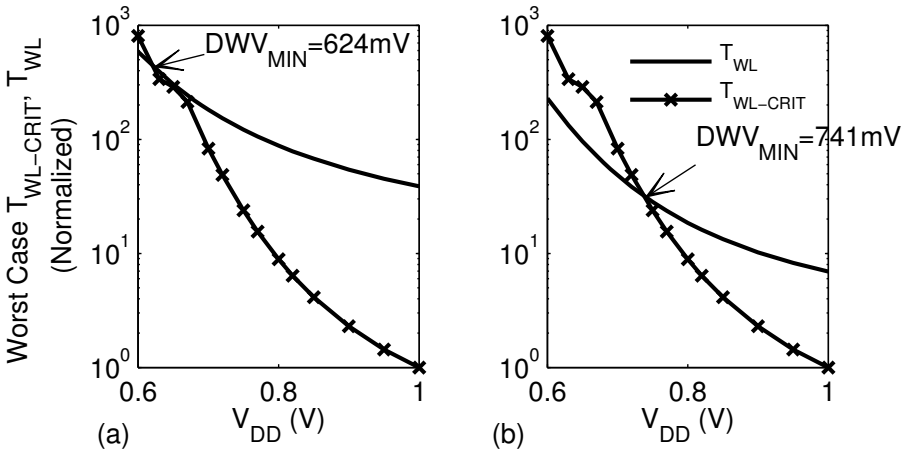


Figure 5.4: DWV_{MIN} dependence on voltage scaling of T_{WL}. DWV_{MIN} increases from 624 mV (a) to 741 mV (b) for two different T_{WL} scaling approaches.

5.4.2 Memory size

As the memory size increases, so does the worst case $T_{WL-CRIT}$ as it moves further out the tail. Fig. 5.5 shows the the variability of $T_{WL-CRIT}$ in terms of the ratio of its worst case and nominal values for various array sizes. The values for the 100kb and 10Mb arrays are obtained using the Statistical Blockade tool (SB), while those for the 1kb and 5kb arrays are obtained from full MC simulation. The worst case $T_{WL-CRIT}$ for a 10Mb array is nearly 120 times the nominal value at 0.8V, while it is only about 20 times the nominal value at 1V. So, for smaller arrays in the order of hundreds of kilobits, the variability impact at lower

voltages is not so severe. However, it is quite significant for megabit-sized arrays.

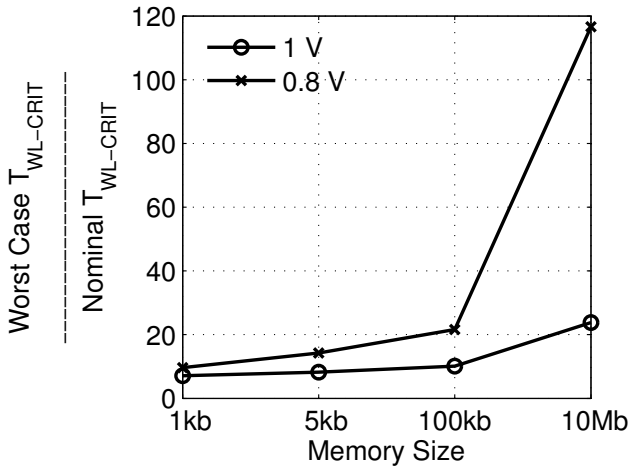


Figure 5.5: Impact of variability on $T_{WL-CRIT}$ for different array sizes.

Thus, as memory size increases, the array becomes dynamic write limited at a higher voltage. For instance, Fig. 5.6 shows worst case $T_{WL-CRIT}$ values across voltage for the 1kb and 5kb arrays. We can observe that the variability is higher at lower voltages and the difference between the worst case $T_{WL-CRIT}$ for the two arrays becomes much larger. As a result, the DWV_{MIN} for the 5kb array is 714 mV, when compared to 624 mV for the 1kb array. Using SB, we determined the DWV_{MIN} for 100kb and 10Mb as well (Fig. 5.7). The DWV_{MIN} for the 5kb and 100kb arrays are almost the same as the latter is static write limited. Due to this, the worst case cell in the 100kb array is relatively stronger than that in the 5kb one from a dynamic writability perspective,as explained in section 5.3.

5.4.3 Bitcell parasitics

Since $T_{WL-CRIT}$ is a dynamic measure of writability, it is affected by the parasitic capacitances in the bitcell, specifically, the capacitances between the storage nodes and

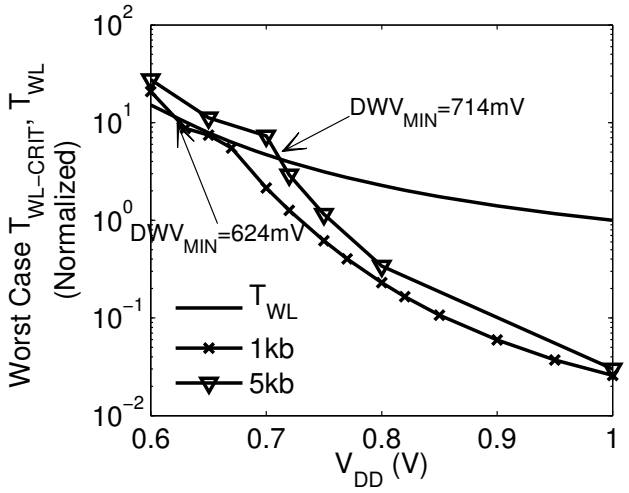


Figure 5.6: $\mathrm{DWV_{MIN}}$ dependence on array capacity. $\mathrm{DWV_{MIN}}$ increases from 624 mV for a 1kb array to 714 mV for a 5kb array.

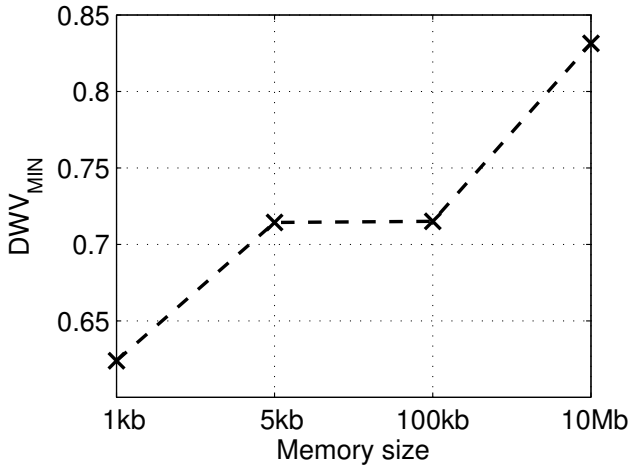


Figure 5.7: $\mathrm{DWV_{MIN}}$ for various array sizes using SB.

various terminals of the bitcell. From the extracted netlist of the bitcell, we note that the inter-storage node parasitic capacitance ($\mathrm{C_{Q-QB}}$) dominates over other components, being at least 2$\times$ larger than the others (Fig. 5.8). Since the storage nodes need to move in the opposite direction for the cell to flip, a larger value of $\mathrm{C_{Q-QB}}$ would make this harder. Thus, $\mathrm{T_{WL-CRIT}}$ is most affected by this component of the bitcell parasitics. As Fig. 5.9 shows,

$T_{WL-CRIT}$ increases by more than 6x if the inter-storage parasitic capacitance increases 10x, with the other components remaining the same.

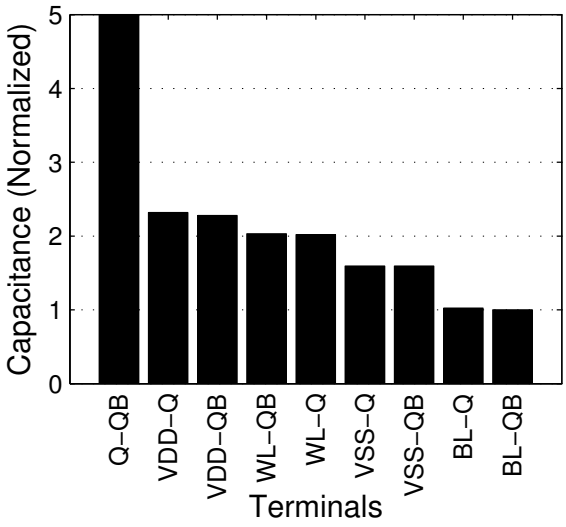


Figure 5.8: Dominant bitcell parasitics.

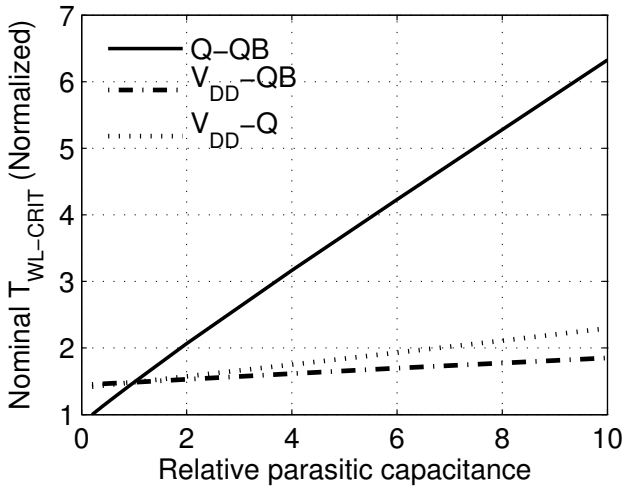


Figure 5.9: Impact of inter-storage node parasitic on $T_{WL-CRIT}$ for each of the three most dominant capacitances, with the others kept constant.

The dependence of $T_{WL-CRIT}$ on bitcell parasitics implies that the DWV_{MIN} also depends on them. Fig. 5.10 depicts the worst case $T_{WL-CRIT}$ across voltage for a 1kb array

with the $T_{WL-CRIT}$ estimated using extracted ("real") and non-extracted ("ideal") versions of the bitcells. We note that while the DWV_{MIN} of the 1kb array with the "real" bitcells is 624 mV, the array with the "ideal" bitcells is dynamically write stable above 600 mV. Again, the number of static write failures is the same in either case. Thus, a layout that reduces the bitcell parasitic capacitances, in particular, C_{Q-QB} can improve the DWV_{MIN} of the SRAM, although, the cell becomes more susceptible to read upsets if C_{Q-QB} is too low. The impact of the parasitic capacitance on the DWV_{MIN} in Fig. 5.10 is small possibly because the cell layout was done carefully to minimize the dominant parasitic capacitances.

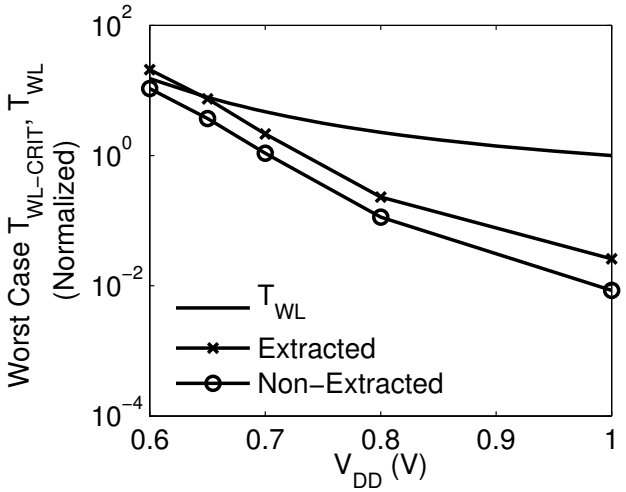


Figure 5.10: DWV_{MIN} dependence on the bitcell parasitics.

5.4.4 Number of Cycles Prior to First Read

So far we have assumed the write to be successful if the cell nodes have completely flipped after a long period of time (e.g. 2-3 orders of magnitude compared to T_{WL}). This definition is highly optimistic as it does not consider the possibility of the cell being read in one of the subsequent cycles. If the cell has not fully flipped before the start of the read,

it is possible that the correct value is not read.

The strictest (e.g. most pessimistic) criterion for write success requires the cell nodes to completely flip by the end of the current write cycle. We call this $T_{WL-CRIT-CYC}$. These definitions form bounds on the real $T_{WL-CRIT}$ of the cell, which thus depends on when the cell nodes are checked to see if they have flipped. The stricter the failure criterion, the larger the $T_{WL-CRIT}$ for a successful write, and the higher the DWV_{MIN}. The $T_{WL-CRIT}$ values defined based on the two write failure criteria described here form lower and upper bounds respecively for the actual $T_{WL-CRIT}$ of the bitcell. The following example shows the impact of the write failure criterion on the $T_{WL-CRIT}$ and the DWV_{MIN}.

Fig. 5.11 shows the nominal $T_{WL-CRIT}$ at two voltages. In either case, the nominal $T_{WL-CRIT}$ initially falls drastically. For instance, if the failure criterion is relaxed by just two cycles, the nominal $T_{WL-CRIT}$ is nearly halved. It eventually settles to the lower bound indicated by the dashed line, where the cell nodes are checked three orders of magnitude after the end of the write cycle. This happens once the failure criterion is relaxed to the point where the cell needs to be first read about 30 cycles after the write cycle.

From Fig. 5.11, we can see how $T_{WL-CRIT}$ depends on the number of cycles prior to the first read operation. Thus if a stricter failure criterion is imposed, the array will be dynamically write limited at a higher voltage. Fig. 5.12 shows the worst case $T_{WL-CRIT}$ across voltage for the two bounds on the write failure criterion — a read immediately following a write and a read after a "long" time (3 orders of magnitude more compared to T_{WL} in this example). We observe that the DWV_{MIN} of the array lies between 624 mV and 744 mV for the two extreme cases of the write failure criteron.

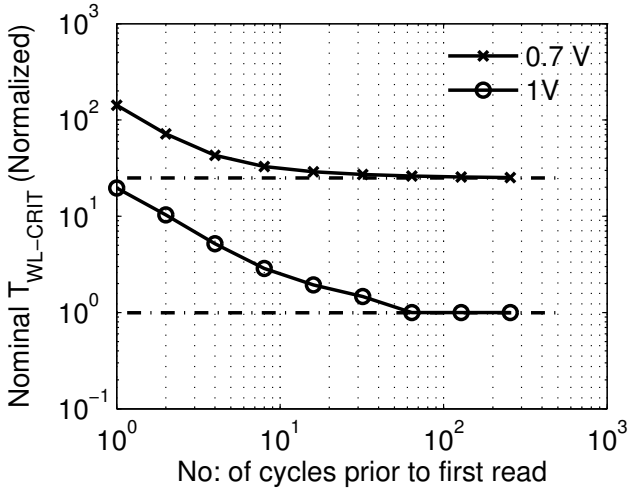


Figure 5.11: Effect of the no. of cycles elapsed before the first read.

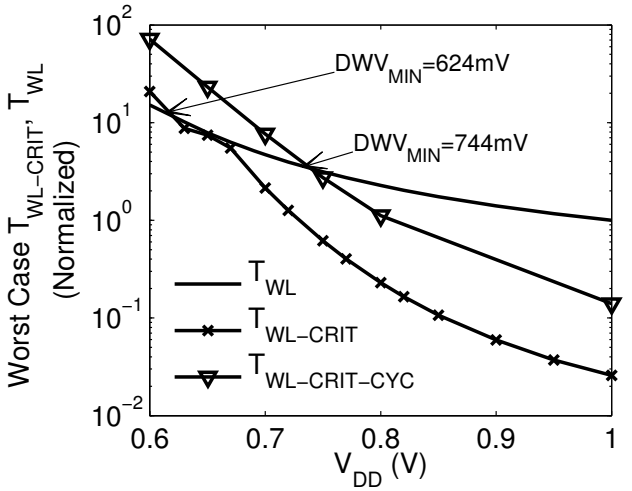


Figure 5.12: $\mathrm{DWV_{MIN}}$ dependence on the number of cycles prior to first read. $\mathrm{DWV_{MIN}}$ for a 1kb array lies between 624 mV and 744 mV.

5.5 Comparison with Static $\mathrm{V_{MIN}}$

In addition to parametric variation, which impacts both dynamic and static limited $\mathrm{V_{MIN}}$, dynamic writability is affected by additional factors, as discussed in the previous secion. In particular, factors such as the voltage scaling of $\mathrm{T_{WL}}$ and the number of cycles prior to first read can influence whether an SRAM array hits the dynamic writability limit

before or after the variability-influenced static write limit is encountered.

Fig. 5.13 shows the static and dynamic write V_{MIN} for various memory sizes. The static V_{MIN} is determined by the voltage at which the first static write fail occurs (e.g. the cell does not flip for any value of T_{WL}). The dynamic V_{MIN} is calculated for the two scenarios of T_{WL} scaling shown in Fig. 5.4. We observe that the dynamic and static write V_{MIN} are comparable when the T_{WL} scaling is heavily margined as in the first case. However, if the T_{WL} scaling is more aggressive, the array hits the dynamic write limitation before static fails start to appear. This is particularly true for large memories in the order of megabits. For instance, when using a more aggressively scaled T_{WL}, the 10Mb memory is dynamic write limited as high as 0.95 V, while static write fails appear only from 0.8 V. Thus, we conclude that for large memories with aggressive performance requirements, dynamic write limitations imposed by the mode of access and T_{WL} scaling will become more dominant than purely variability affected static write limitations.

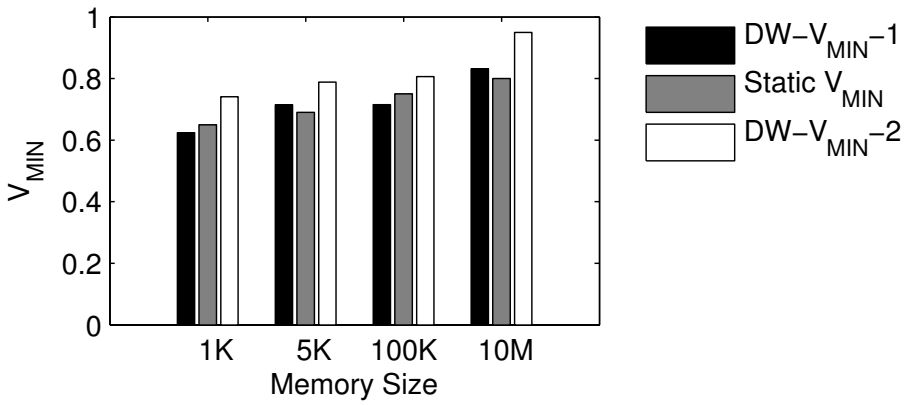


Figure 5.13: Dynamic vs. Static V_{MIN} for self-timing path generated, heavily margined T_{WL} (a) and aggressive bitline differential dependent T_{WL} (b).

5.6 Impact of Write Assists

Several implementations of write assists exist in literature. Voltage bias-based write assists fall broadly into two categories — ones that alter the "noise source" amplitude or duration through the access transistor, and ones that modify the strength or voltage transfer characteristics of the cross-coupled inverters [15]. We choose the WL boost method (WLB) from the former and V_{DD} lowering method (VDDL) from the latter categories, as these appear to be the most popularly used write assist methods in recent literature [50][56][57]. We use a WL boost and V_{DD} droop value of 100mV.

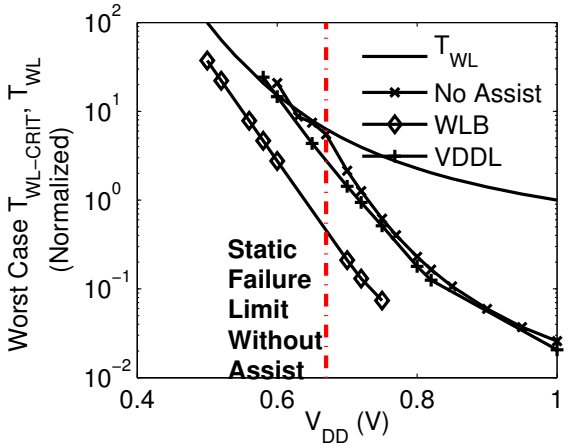


Figure 5.14: Impact of write assists on worst case $T_{WL-CRIT}$. Static write failure occurs without assist at 670 mV. No static write failures are observed above 500 mV with either assist.

Fig. 5.14 shows the worst case $T_{WL-CRIT}$ in a 1kb array, with no write assists, and with WLB and VDDL. The T_{WL} in these examples corresponds to a WL pulse generated from a self-timed path. We observe that WLB is more effective than VDDL in reducing the worst case $T_{WL-CRIT}$. This is because the gate-to-source voltage of the access transistor is higher in the case of WLB and consequently, the write time is lower, leading to a lower worst case $T_{WL-CRIT}$ and DWV_{MIN} . However, both categories of assist techniques eliminate the static

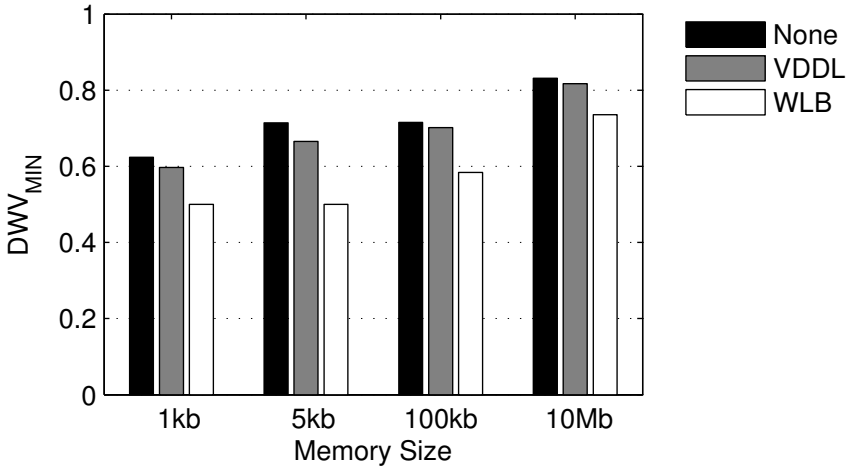


Figure 5.15: $\mathrm{DWV_{MIN}}$ for various array sizes with and without write assists. Static $\mathrm{V_{MIN}} < \mathrm{DWV_{MIN}}$ for both assist methods.

failures that appear when no write assist is applied, down to 500 mV. In this case, WLB is a better write assist purely from a $\mathrm{T_{WL-CRIT}}$ point of view, although it worsens the stability of the half-selected cells during read. Fig. 5.15 confirms this for larger memories as well. This agrees well with the conclusions drawn in [15] and [53] about the best write assist techniques, based on static and dynamic writability considerations respectively. In terms of half-select stability however, VDDL is a better write assist, particularly if the $\mathrm{V_{DD}}$ is shared column-wise since the boosted WL voltage reduces the read stability of the half-selected cells along the same row. We note that other considerations such as power overhead and complexity of implementation would also need to be considered before choosing a write assist implementation.

5.7 Conclusions

We have discussed how bitcell writability can be described from a dynamic standpoint through the $\mathrm{T_{WL-CRIT}}$ metric and introduced the concept of $\mathrm{DWV_{MIN}}$ that is distinct from

the operating SRAM V_{MIN} defined based on conventional SNM. Though more accurate, dynamic metrics are more difficult and expensive to compute. The concepts presented in this dissertation can potentially be investigated further to find an explicit relationship between static and dynamic writability metrics or conventional V_{MIN} and DWV_{MIN}. For instance, we have shown how certain factors affect the DWV_{MIN} but not the conventional V_{MIN}. Using such information, if a relation between dynamic and static metrics can be determined, then by simply performing simpler, less expensive static writability analysis, we can determine a more accurate operating SRAM V_{MIN}. This V_{MIN} would take into account the dynamics of the write operation and not be solely based on the DC characteristics of the write operation.

Chapter 6

Virtual Prototyping Tool

6.1 Motivation

While process scaling has enabled ever-larger embedded memories, scaling trends such as process variability, device leakage, soft error susceptibility, and interconnect delay make memory design increasingly difficult. In the face of such scaling effects, the best way to design SRAMs that are optimal in terms of global figures of merit (FoMs), which we define as energy, performance, area, and yield, at the 32nm process technology node and below, largely remains an open question. Researchers have proposed a number of techniques at the technology and circuit level to deal with problems such as variation and leakage [47][58][59][60]. The alternative bitcells and pseudo-differential sensing proposed in the previous chapters are also examples of circuit techniques to deal with various challenges of SRAM design at advance technology nodes. However, all these techniques tend to address only certain individual components of the memory. A change in any one of the key memory circuits will alter the optimal circuit topologies, array partitioning, and

architecture for the entire memory. This makes it difficult to evaluate a particular circuit technique without assessing global benefits and overheads.

Back-of-the-envelope estimation of overheads and impact on SRAM global FoMs early in the design flow tends to be ad-hoc and dependent on assumptions that vary from designer to designer. Alternatively, complete SRAM prototypes to evaluate each new technique can be created. However, this impractically increases design time and reduces productivity. Thus, there is a need for a methodology through which designers can generate and evaluate prototypes at every step of the SRAM design process that account for process and circuit level issues in terms of global FoMs. Thus, there is a need for a methodology through which designers can generate and evaluate prototypes at every step of the SRAM design process that account for process and circuit level issues in terms of global FoMs.

To address this problem, we present a Virtual Prototyping tool (ViPro), which enables early design space exploration by creating virtual prototypes of a complete SRAM macro, even when many design details are missing (hence virtual). As the design process proceeds, the prototypes become more accurate and complete. Thus, ViPro helps the designer do what he would want to do anyway (e.g. design space exploration), but much more efficiently, making it design automation in the truest sense. ViPro has four key features that make it a valuable tool for SRAM designers. First, it generates a base-case SRAM prototype. Second, since the model allows the components to be described using varying levels of detail, designers can define and work on a complete prototype quite early in the design cycle. Third, it can quickly re-optimize the design if a circuit component or the process models change. Finally, it performs its own process characterization, and thus can be used with any process with defined SPICE/Spectre device models.

The novelty of the tool is that it generates a virtual prototype of the SRAM with as much information as is available at every stage of the design process, and gives the best possible estimate of the global FoMs and trade-offs between them without having to perform full-fledged circuit simulations. It is not a stand-alone tool or a compiler that designs an SRAM. Rather, the designer is actively involved, and uses his expertise to make design decisions based on the output of ViPro, which ultimately leads to an optimal design after several iterations. The estimates provided by ViPro may be inaccurate early on, but as the design proceeds and more components are clearly defined, the accuracy of the prototypes improves.

6.2 Related Work

There are a few memory design and FoM characterization tools available, but they do not support integrated process-circuit-system co-design like ViPro. At one end of the spectrum are architecture-level modeling tools like CACTI [61], used by computer architects to obtain quick estimates of SRAM access time, power, and area. CACTI 6.0 [62] facilitates high level design space exploration by using an optimization cost function that accounts for a user-weighted combination of delay, leakage, dynamic power, cycle time and area. Our tool also supports architectural exploration, but it differs from CACTI in two key ways. First, CACTI makes fixed assumptions regarding the circuits comprising the SRAM, so it optimizes at the architecture level only. ViPro allows designers to generate circuit information (via simulation) specific to any given technology or to add/alter the underlying circuits. Thus, it supports circuit-architecture co-design, which leads to better overall designs. Second, CACTI supports a limited set of process technologies and assumes ITRS [63] parame-

ters for all its calculations. These assumptions may not be accurate, especially for advanced process nodes. ViPro uses a technology-agnostic simulation environment (section 6.4) to characterize its circuit components in any process using Spectre simulations before generating the virtual prototypes, so it uses accurate technology-specific circuit parameters for any process.

At the other end of the spectrum are transistor-level optimizers (e.g., [64][65]) that are good at choosing optimal device characteristics (e.g. W/L, V_T, V_{DD} etc.) for a given circuit topology, but are not helpful in choosing an optimal circuit topology or micro-architecture. In addition, since they are designed to thoroughly explore the design space under device and environmental variations, they are not suitable for quick, early design space exploration.

Finally, memory compilers (e.g., [66][67]) help generate memories based on user-defined parameters, but do not provide trade-off information or facilitate design space exploration and optimization. They are more a deliverable from memory design teams rather than a tool for memory design teams. ViPro fills the gap between these tools by providing an optimization and design space exploration tool for SRAM that supports circuit-system co-design. The Venn diagram in Fig. 6.1 succintly sums up the role of ViPro in SRAM design.

An analogous design tool for the design and optimization of high speed links was previously presented in [68], which couples circuit level parameters with global FoMs and demonstrates the value of doing so for exploring a broad design space. The tool relies heavily on analytical expressions that are specific to high speed links. ViPro targets another complicated mixed signal design problem, SRAM, with a more generalized approach. While it can support analytical modeling for speeding up the optimization process, it also

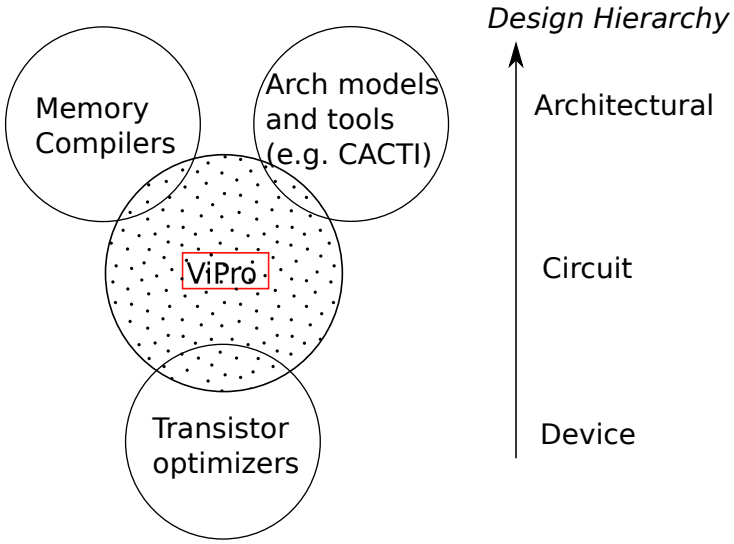


Figure 6.1: Although there is some overlap between the functionality of ViPro and existing SRAM design tools, the novelty of ViPro is that it is the first such tool that fills the gaps between architectural simulators, transistor optimizers, and memory compilers.

supports a fully simulation based approach or a mixed modeling/simulation framework (e.g. by supporting variable levels of detail in describing the SRAM circuit components).

6.3 Overview

Fig. 6.2 shows the structure of ViPro, which comprises two main blocks – a technology-agnostic simulation environment (TASE), and a hierarchical metacompiler (HMC, meta to distinguish it from a true compiler that produces complete final designs). TASE provides a technology-agnostic framework for generating data from simulation of SRAM components, so that ViPro can operate in any process technology. The HMC implements an editable and flexible hierarchical model of the memory (HMM) that allows a designer to define components of the memory with varying levels of detail and accuracy, performs optimization, and generates schematics and layout of the SRAM and its component circuits.

Optimal bitcell schematic and layout are typically provided by the foundry. Alternatively, the designer can use the bitcell generator (section 6.5) to generate a logic-rule bitcell [69].

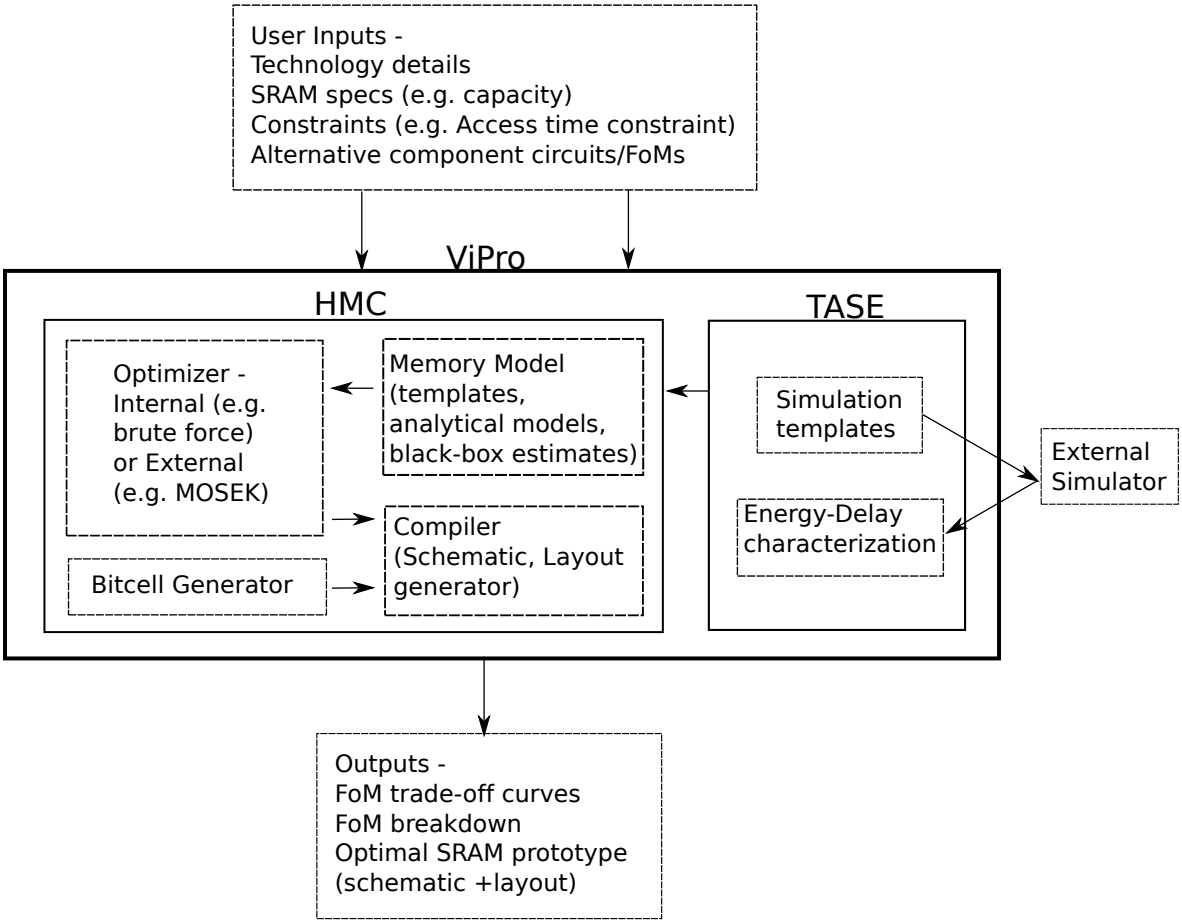


Figure 6.2: Structure of ViPro showing various components, and inputs and outputs for the tool.

The designer provides the following inputs in a user configuration file.

- Process technology
- Top-level memory specifications capacity, word-size, supply voltage, operating temperature, etc.
- Constraints on metrics like energy and delay

- Component specifications (optional) black-box estimates, analytical models or TASE simulation templates

ViPro produces the following outputs.

- Trade-off curves between global FoMs (e.g. E-D curves)
- Global FoMs for a fixed user-specified SRAM configuration (no optimization performed)
- Break-down of FoMs among components
- Schematic and Layout of SRAM component circuits (e.g. WL drivers, SA etc.)

6.3.1 Components

TASE abstracts out the process dependencies from the simulation set-up using simulation templates. It then combines these templates with process-related data to produce the simulation-ready netlist and simulates it using Spectre. We incorporate TASE into ViPro for two purposes. First, to characterize a technology or process through device-level simulations (e.g. I-V curves, FO4 delay, etc.). Second, TASE includes a user-expandable library of templates of SRAM components that can be characterized in terms of global (e.g. energy and delay) and component-specific FoMs (e.g. noise margins for a bitcell and offset for a sense amplifier). These templates also use Monte Carlo analysis to capture statistical data required for yield estimation. TASE can also be used stand-alone as a simulation platform for any kind of design, as described in section 6.4.

The HMC can be broken down into four parts. First, the bitcell generator (section 6.5) can be used to generate an optimal logic-rule compliant 6T bitcell based on considerations of area, RSNM, WNM, and I_{READ}. Alternatively, this block can be unused if the

foundry provided "pushed" rule 6T bitcell is used instead. Second, a hierarchical model of the SRAM implemented in MATLAB (section 6.6) determines global FoMs using data from TASE. The model can make this calculation using inputs with variable level of detail, which makes it an important part of ViPro. Third, an optimizer (section 6.7) determines the values of the design knobs (e.g. number of rows, columns, device sizes etc.) to meet user-specificed constraints and requirements for the global FoMs. Fourth, the actual compiler (e.g. schematic and layout generator) that generates the schematic and layout for the SRAM for any technology. The technology-agnostic nature of TASE and the SRAM generator described in section 6.8 makes ViPro usable for any technology.

6.3.2 Design Methodology

The flowchart in Fig. 6.3 depcits the steps involved in using ViPro for generating virtual prototypes. The first step (once per technology) is to characterize the process technology through device-level simulations. In the second step, TASE characterizes components from the library (once per technology). Any components currently unavailable in the TASE template library must be defined using black-box estimates, analytical models, or new templates added to the library. As more components are added to the library, the accuracy and scope of the virtual prototypes improves. In the third step, using existing components and built-in analytical models, ViPro can generate an optimized base-case prototype that provides a convenient starting point for a designer interested in creating a more optimized custom design. He can explore different circuit options (e.g. assist techniques, alternative bitcells, etc.) to further optimize his design. By changing the specification of one or more components and running the tool iteratively, the designer can steer the design effort

towards an optimal design. Alternatively, the designer can exploit the tools technology-agnostic nature to compare prototypes for different process or device options. This kind of comparison is especially useful for many fab-less companies that have to choose between different processes for their design. Thus, for example, when porting an existing design to a new technology, the designer can quickly see how the optimum configuration of the design changes. This technology agnostic feature also allows for process-circuit co-design, since the optimal circuit and architecture selections will change in response to process alterations.

A key insight here is that the designer is an integral part of the tool flow. As the designer runs the tool multiple times and compares several virtual prototypes, his understanding of the trade-offs involved in the design increases. Thus, he finalizes more and more components, which improves the accuracy of the prototype. Ultimately, as the design nears completion, all the components in the memory are specified in terms of circuit netlists from the template library of TASE, and our tool becomes closer to an actual SRAM compiler in terms of generating full schematics and layout.

6.4 TASE

In this section, we describe the implementation of TASE and its usage methodology. We start with a description of the simulation templates and execution files. Then, we describe the tool flow. Finally, we demonstrate how TASE can be used both as a standalone tool, and as a part of ViPro.

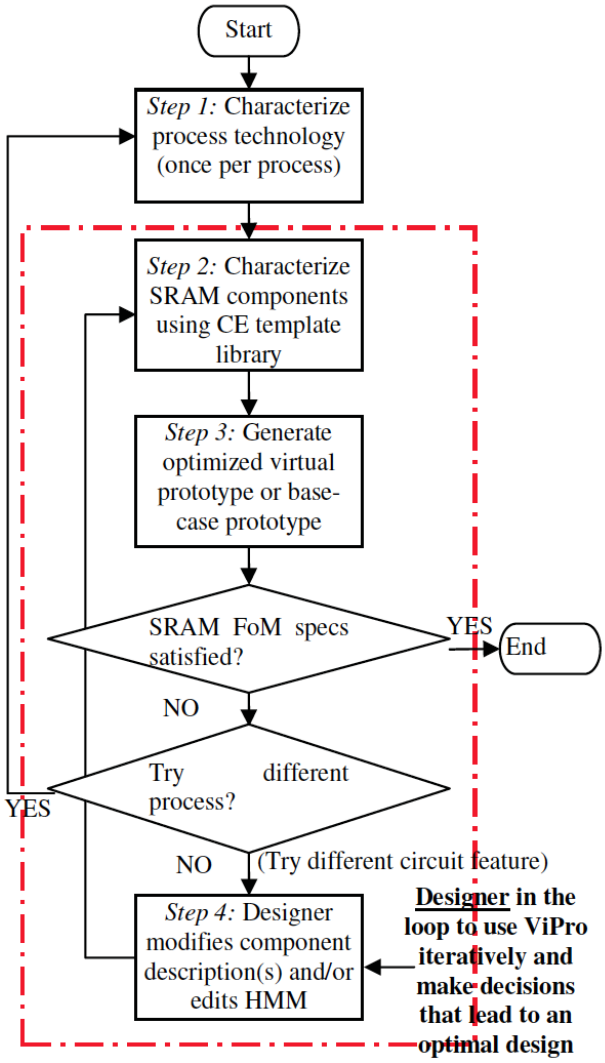


Figure 6.3: Methodology of using virtual prototypes for SRAM design.

6.4.1 Tool Overview

TASE acts as a wrapper, currently implemented in MATLAB and PERL, to an existing SPICE-like simulation framework (e.g. Spectre). TASE is built around a low-level circuit simulator, and is best suited for quick characterization of relatively small circuit blocks composed of a few tens of transistors. In order for TASE to perform technology agnostic

simulation, it must cleanly separate the technology-dependent aspects of a simulation from the technology-independent aspects. It is these technology-independent aspects that can propagate designer knowledge and intent. This is the key insight that enables TASE to reuse simulations for any technology or process corner.

The input to circuit simulators such as SPICE or Spectre has the following basic components, all of which contain technology-specific information.

- Netlist circuit topology description
- Stimuli and initial conditions
- Analysis statements (e.g. Transient, DC)
- Measurement statements (e.g. Delay, Power)
- Temperature and voltage of operation
- Device models

We separate the components of a TASE simulation into a suite of technology independent templates and a single technology-specific configuration per process. A template exists for each simulation, and each technology agnostic template contains several components (Fig 6.4(a)):

- Technology agnostic netlist
- Technology agnostic stimuli and initial conditions
- Technology agnostic analysis and measure statements
- Technology agnostic post-processing script

First, the netlist uses technology-independent references to subcircuits that represent FETs (“P_TRANSISTOR” and “N_TRANSISTOR”). Device sizes, etc., are not specified

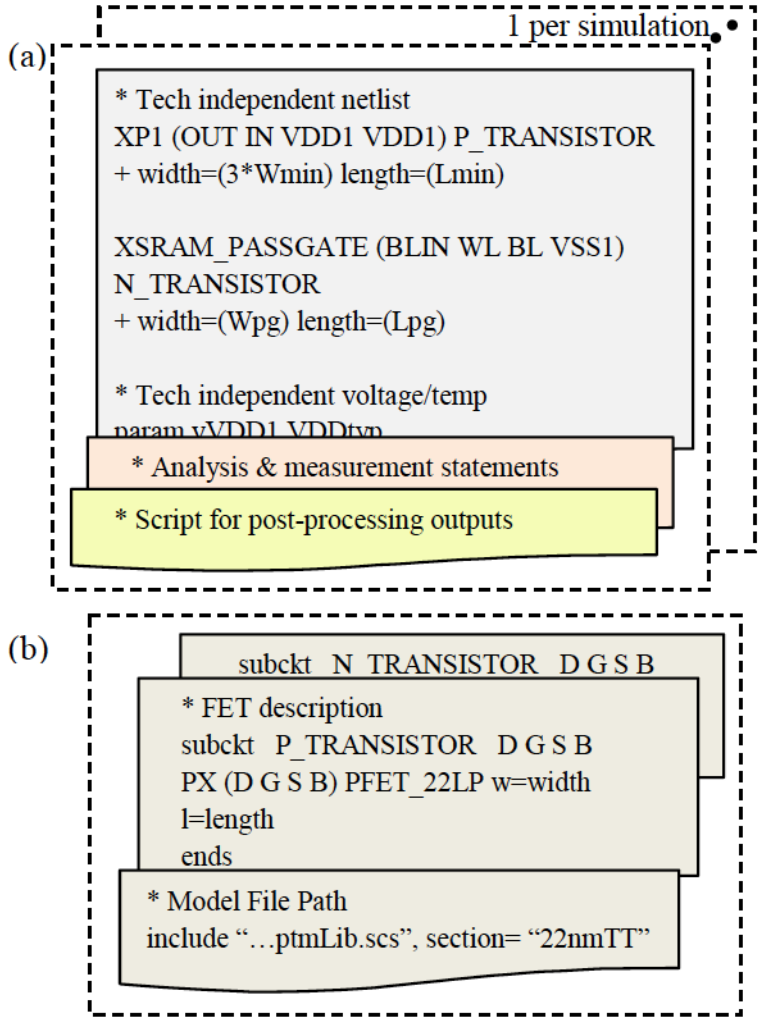


Figure 6.4: Example (a) technology agnostic template (b) technology specific configuration.

using absolute numbers, but in terms of minimum feature size or as parameters to be defined by the user. Second, analysis and measurement statements set up the simulation and capture the relevant results. Many features of these statements are already process independent, but values that must change with process (e.g. transient run time) are parameterized. Finally, optional post-processing scripts (e.g. MATLAB) extract, process, and plot data from the simulation results. The technology-specific configuration (Fig. 6.4(b)) contains all of the

information that changes with each of the different processes:

- Device model files
- Temperature and voltage of operation
- Simulation parameters (e.g. Monte Carlo seed)
- Template parameters (e.g. device sizes)

Technology specific sub-circuits for NMOS and PMOS devices, with links to the device models, provide the generic P_TRANSISTOR and N_TRANSISTOR components used in the simulation templates. The structure of TASE makes it quite easy to add new content. To add a new simulation, the designer must simply build a new technology independent template and place it in the templates directory, then add definitions for any new parameters to the top level execution file. Building a template is only slightly more time consuming than writing a direct simulation, so the overhead of this procedure is justified since the template can be re-run in any technology without modification. Adding a new technology requires only writing a new technology-specific configuration, which contains only the transistor sub-circuits and model path information. Now, these analyses and insights into the tradeoffs that lead to certain design decisions will port with the final design into new processes.

6.4.2 Tool flow

To run TASE, the user sets up a top level execution file (Fig. 6.5 shows an example execution file) that can associate multiple templates into a group of related simulations. Through the execution file, the designer selects a technology-specific configuration (e.g. sets the netlist parameters and model files) and a group of templates to run, which can be

selected individually or for a particular category (e.g. process characterization, SRAM, etc.). Currently, TASE supports templates in Spectre or Ocean formats.

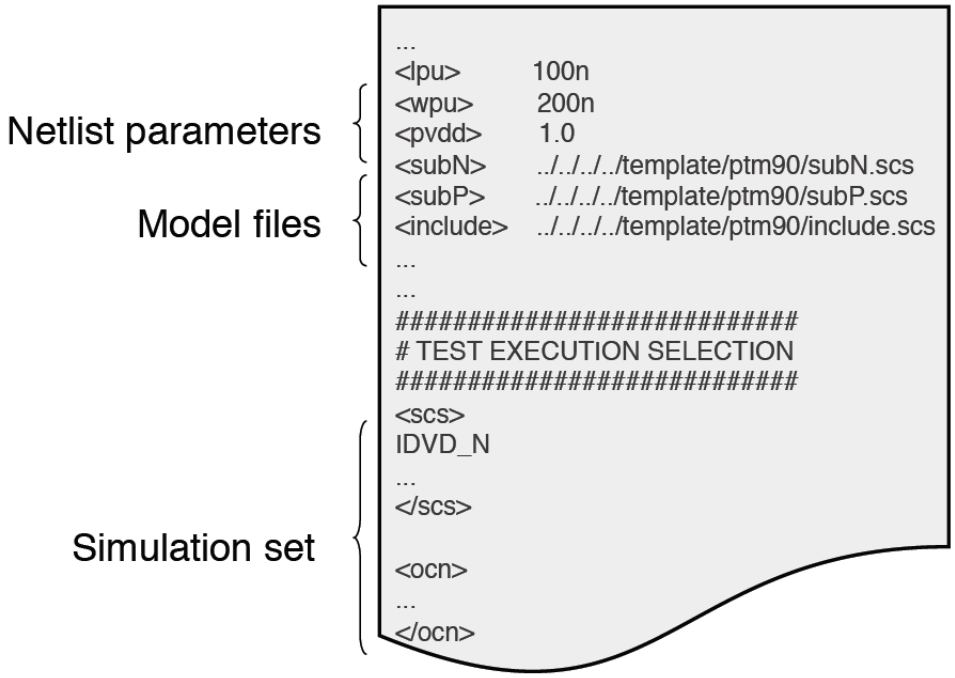


Figure 6.5: Example execution file for TASE.

Fig. 6.6 depicts the TASE tool flow. TASE combines the parameter values from the execution file and the specified model-file with the simulation templates, which converts the technology agnostic templates into simulations customized for the chosen technology. Next, TASE launches the circuit-level simulator(s) to run the selected simulations. After they conclude, TASE consolidates the post-processing scripts (MATLAB) from the templates, launches the consolidated script to process the raw output data from the multiple simulations, and produces plots and results for the designer. Since templates can be reused in multiple groups, the post-processing for a given template can be specifically tailored to the context of the simulation in each group. This also supports using TASE as an iterative

design tool, since simulation templates can be reused and refined in the context of a new execution file.

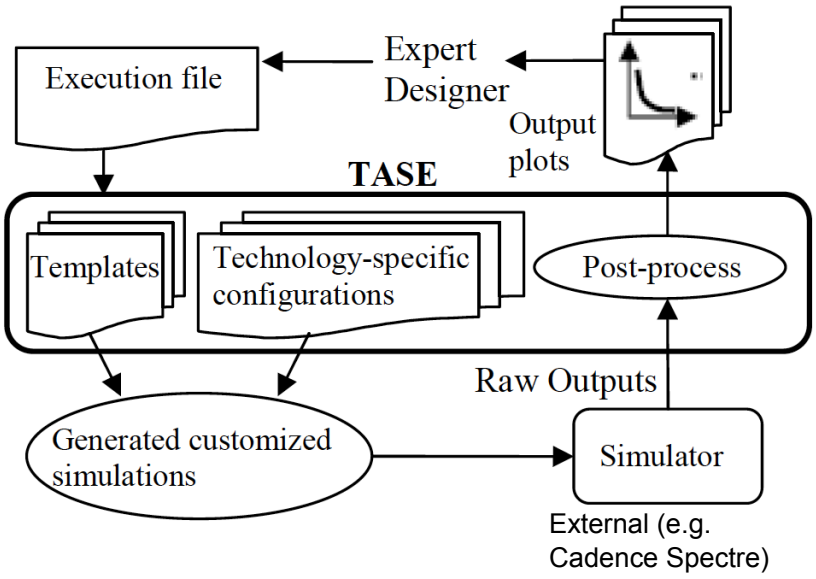


Figure 6.6: Tool flow diagram for TASE.

6.4.3 Usage

As a standalone tool

The technology agnostic nature of TASE makes it a useful tool for several purposes. First, it can be used for exploring a new process by running existing templace for device characterization with an execution file for the new process. Second, it can help forecast the behavior of any circuit (e.g. digital, analog, mixed-signal) or device using Predictive Technology Models (PTMs) [70]. Finally, by simply adjusting parameters in the execution file, TASE enables quick simulation across different process corners, voltages, and temperatures. Reference [71] describes several examples that show how TASE can be useful as a

standalone tool.

As a part of ViPro

TASE is used to characterize SRAM component circuits (WL driver, timing block, decoders, column-mux, SA, and write drivers) in terms of energy and delay with respect to various knobs such as device sizes, number of stages in buffer chains, aspect ratio of the array etc. This data is then used by the Optimizer to determine optimal values for these knobs to generate an SRAM that meets the top-level FoM constraints.

6.5 Bitcell Generator

The area of the SRAM and the parametric yield are primarily affected by the bitcell, especially for larger memories. This allows us to optimize the bitcell independent of the remainder of the SRAM. The bitcell generator module of ViPro generates an optimal bitcell design based on area, I_{READ}, and stability (e.g. RSNM, WNM) constraints. It is common for design kits to come with an already optimized SRAM bitcell or array provided by the foundry. In that case, the designer can directly use the provided bitcell instead of using ViPro to generate it.

The optimal bitcell design is determined as follows. The designer provides search windows and step sizes for the bitcell device sizes, and constraints on the area, margins, and I_{READ} as input to the bitcell generator. First, the bitcell area is calculated based on an area model that supports both logic-rule compliant and pushed rule bitcells. The model parameters are set based on a commercial 45 nm technology and extrapolated for other technologies. Next, bitcells that do not satisfy the area constraint are filtered out. Then,

simulations are run in TASE to filter out bitcells that do not satisfy the remaining constraints. Finally, we are left with a small subset of bitcells from the original search space that satisfy the area and metric constraints. The cell with the smallest area is chosen as the final bitcell. Fig. 6.7 captures the working of the cell generator module.

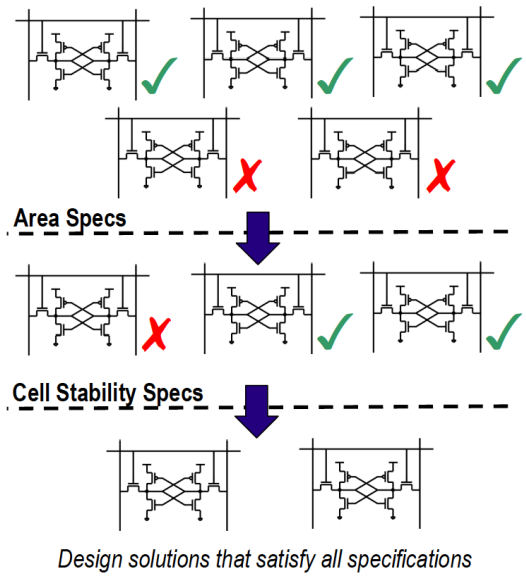


Figure 6.7: Optimal bitcell design through search space reduction.

6.6 SRAM Model

ViPro contains a hierarchical model of the memory implemented using object-oriented MATLAB. Fig. 6.8 depicts the memory architecture currently implemented in the tool. For the ease of initial demonstration, we use only two levels of hierarchy and support low-capacity (e.g. smaller than 1 Mb). The components of the memory are considered to be the following:

- Column-mux including the BL precharge and transmission-gates (CD)

- Sense amplifier and associated precharge and output latch (SA)
- Write drivers and read flip-flop (IO)
- Bitcell array (BC)
- WL Drivers (WLD)
- Predecoder, column-decoder, and timing block (TMG)

The SRAM model first calculates the FoMs for each component. The global FoMs are then calculated using the component FoMs. For instance, the total energy is calculated by simply adding the energy of the component circuits and the delay by adding the delays of the components on the critical path.

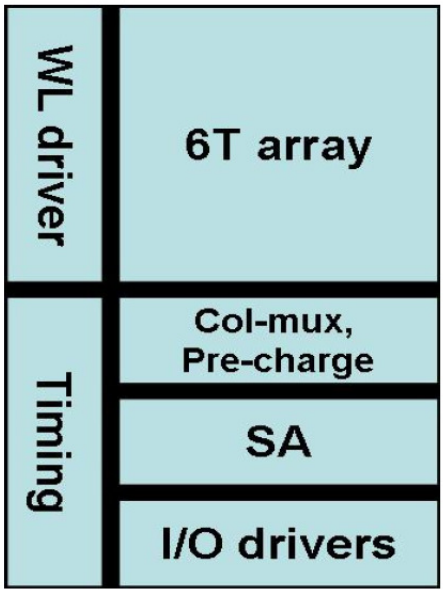


Figure 6.8: SRAM architecture and hierarchy assumed by ViPro.

Each component in the SRAM hierarchy is defined as a class, and each class has properties defined for parameters (device sizes, voltages, etc.) and FoMs associated with the component. The methods (functions) in the class that determine the values for an instance's

FoMs support varying levels of detail. For example, the method that computes delay can simply assign it a constant value, equivalent to treating the component as a black box with estimated constant behavior. This may be the best option for a new circuit block about which little details are known. The method can alternatively use an analytical expression or macro-model to compute the FoMs from the components parameters. For instance, the energy of a wordline (WL) driver is estimated using CV^2 calculations, with the value of C determined by the bitcell pass-gate and WL wire capacitance. The delay of a component is calculated using logical effort or using an equivalent simplified R-C circuit. Finally, the components can be specified by transistor level netlists from which TASE can directly obtain FoM data across different values of the input parameters, leading to complete tradeoff curves.

A key feature of the HMM is the use of class inheritance that allows properties common to a branch of the hierarchy to filter down to the leaf node of that branch. Fig. 6.9 shows the parent class and the SA class that is inherited from the parent class. By defining common parameters whose optimum value is affected by multiple blocks as properties of the parent class, the HMM can capture interactions between different blocks when optimizing the overall design. For example, the offset of the sense amplifier (SA) and the strength of the bitcell's access transistor both influence the optimum number of bitcells on a single bitline (i.e. number of rows (NR) in the memory array). Thus, the HMM defines a property "NR" in the parent SRAM class that filters down to both the bitcell and the SA classes (see Fig. 6.9). The local component methods capture how "NR" depends on both the offset in the SA block and the drive strength of the bitcell block. These local definitions allow top-level optimization to coordinate local dependencies. For instance, the SA can adjust its

(a)

```
classdef SRAM_Component
  properties
        W; % transistor min. width
        L; % transistor min. length
        VDD; % nominal operating voltage
        NR; % Number of rows per block
        ...
  end % end properties

  methods
        function obj = E()
        % use black-box estimates or CE
characterized blocks for energy
        end
        ...
  end % end methods
```

(b)

```
% SA is a sub-class of the parent class
SRAM_Component
classdef SRAM_SA < SRAM_Component
  properties
        offset;
        % NR, by inheritance
        <other attributes local to SA>
  end % end properties

  methods
        function obj = E()
        % overloaded function to calculate
energy for SA component
        end
        ...
  end % end methods
```

Figure 6.9: (a) SRAM parent class and (b) example component class showing circuit attributes and FoM estimation.

offset (e.g. by changing its device sizes) or the bitcell can adjust its drive strength (e.g. by strengthening its access transistor).

6.6.1 Model Verification

We verified the SRAM model for a 65 nm commercial bulk CMOS technology in the following way. We generated the full SRAM schematic of a 512x16 macro with a 16-bit

word-size using ViPro . We simulated it using Cadence ultrasim and determined the average power numbers for each component using the power analysis command in ultrasim. The energy is obtained by multiplying the average power with the duration of the simulation. The delay is measured as the time difference between the 50% points of the two waveforms being measured. Tables 6.1 and 6.2 show the comparison between the energy and delay numbers respectively reported by the ultrasim simulation of the full SRAM and from the SRAM model in ViPro.

Table 6.1: SRAM energy verification with a 512x16 macro

	Write (pJ)			Read (pJ)		
	ViPro	Macro	Error	ViPro	Macro	Error
Decoder	1.16	0.86	+30%	1.16	0.86	+30%
Bitslice	*3.76*	*3.72*	*+1.1%*	*0.76*	*0.86*	*-11.6%*
Array	0.27	0.26	+3.8%	0.4	0.56	-28.6%
Timing	0.66	1.5	-56%	0.66	1.64	-60%
TOTAL	**5.85**	**6.32**	**-7.4%**	**3.01**	**3.7**	**-18.6%**

From these results, we observe that the energy and delay models are not highly accurate for every individual component circuit. We believe these discrepancies are due to two reasons. One, the macro numbers are from ultrasim's power analysis command, while the ViPro numbers are from Spectre simulations. Ultrasim is less accurate when compared to Spectre, even at the slowest speed setting as it compromises on accuracy to some extent to improve simulation speed. Second, the E-D characterization performed by TASE assume

Table 6.2: SRAM delay verification with a 512x16 macro

Write (ps)					Read (ps)		
	ViPro	Macro	Error		ViPro	Macro	Error
Decoder	378	397	-4.8%	Decoder	378	397	-4.8%
Bitcell flip	330	227	+45.3%	Bitline droop	425	330	+28.8%
				SA delay	125	125	0%
TOTAL	**709**	**624**	**+13.6%**		**928**	**852**	**+8.9%**

the control signals (e.g. WL enable, BL precharge enable etc.) to arrive at the same time. However, in the macro this is not the case. The BL precharge enable and the signals that enable the tri-state inverters in the IO drivers arrive a little earlier than the WL. This results in glitches in the BL voltage or short-circuit power leading to an overall higher energy consumption reported by the macro when compared to ViPro's estimates.

Though some individual sub-components show a high error, the overall numbers are much closer. For instance, the write energy numbers estimated by ViPro are only 7.4% different from that calculated from the macro. This is because the dominant energy consumption is due to the BLs, which is estimated quite accurately. The large discrepancies are in components that do not contribute much to the overall energy consumption resulting in less error in the overall energy consumption reported by ViPro.

6.7 SRAM optimization

The SRAM optimization problem is a specific case of circuit optimization that falls under the category of constrained optimization problems. In these problems, we seek to minimize or maximize an objective function (e.g. the global SRAM FoMs in our case), given certain constraints (e.g. performance, area, power etc.). It has been shown that most circuit design problems can be formulated or approximated as a geomeric program (GP) or generalized geometric program (GGP) [72]. A GP or a GGP is characterized by objective and constraint functions that have a special form. These problems can be transformed to a convex optimization problem [73] and then very efficiently solved using commercial tools such as MOSEK [74]. For instance, the Stanford Circuit Optimization Tool (SCOT) uses MOSEK as a plug-in to determine optimal device sizes, V_T, and V_{DD} [75][76] for logic circuits such as buffer chains. A similar solution could potentially be implemented in ViPro to determine circuit (e.g. device sizes) and architecture parameters (e.g. number of rows and columns) that result in optimal FoMs for the SRAM.

In this dissertation, for the ease of demonstration of the virtual prototyping methodology, we limit ourselves to a small number of knobs and perform a brute force search to determine the optimal E-D curves (e.g. section 6.9). Each point on the optimal curve corresponds to a particular combination of the knob values. Depending on the user constraints on the energy or delay of the SRAM macro, ViPro determines the optimal combination of the knob values. Using these values (e.g. number of rows, device sizes etc.), the schematic generator (section 6.8) can generate the SRAM schematic. Currently, ViPro supports the following knobs for optimization.

- Aspect ratio/number of rows of the SRAM Macro

- Number of pre-decode and WL driver buffer stages (to set the optimal decoder)
- Write driver device width

The SRAM optimization problem will only be truly complete if area and yield are also considered along with energy and delay as the FoMs. However, for the purpose of demonstrating the SRAM design methodology using virtual prototyping, considering a subset of the metric suffices.

6.8 Technology Agnostic SRAM Compiler

6.8.1 Schematic Generator

The schematic generator, implemented in SKILL, automatically generates full schematics for the SRAM in any technology using the optimal SRAM parameters determined by ViPro. SKILL is a scripting and parametrized-cell description language used in several Cadence design tools. The generator is technology-agnostic and the user simply needs to point ViPro to the correct library containing the basic devices for the technology and specify the default parameters for the technology, such as minimum device sizes.

The schematic generation works in a hierarchical manner as follows. First, the basic gates and blocks, such as inverters, and gates, and buffer chains are created. These blocks and gates are then used to create the "leaf nodes" of the SRAM (e.g. WL driver, column-mux, SA etc.). The following are considered as the leaf nodes of the SRAM.

- Bitcell Array
- WL Driver
- Bitslice (comprising the column-mux, SA, and I/O circuits and drivers)

- Timing and Predecoder (comprising the control signal generation logic and 3-to-8 predecoders)
- N-chain buffers (the buffer chain driving the predecoded row address output to the row-pitchmatched WL drivers)
- K-chain buffers (the buffer chain driving the WL output to the bitcells)

Finally, the leaf nodes are tiled together to generate the SRAM macro schematic. Fig. 6.10 shows the hierarchical structure of the gates and blocks created by the generator, starting from the basic gates and leading up to the complete SRAM schematic.

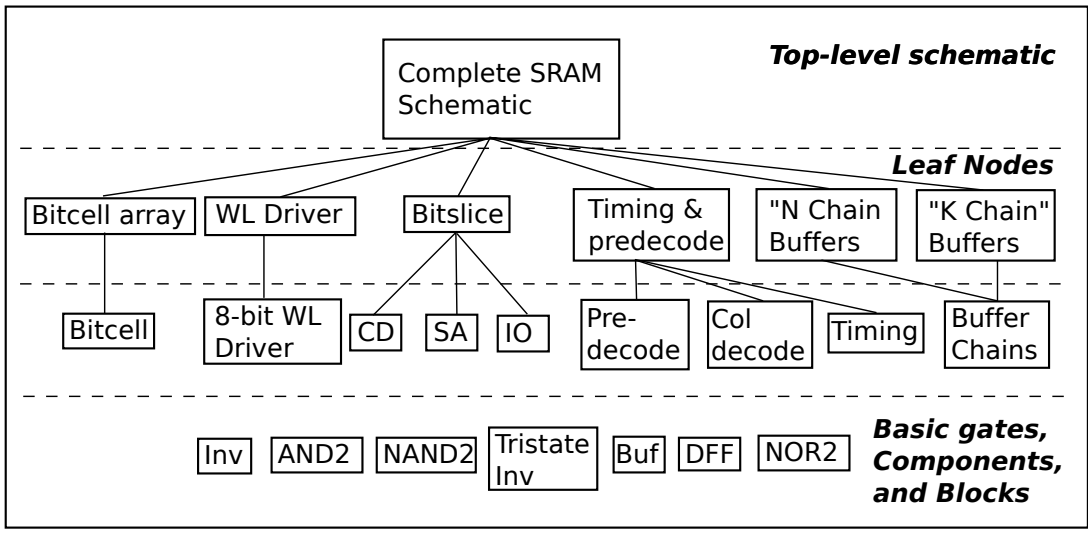


Figure 6.10: Hierarchical block diagram of the generated SRAM schematic.

6.8.2 Layout Generator

ViPro also generates the layout for the SRAM macro. However, unlike the schematic generation which is fully automatic starting from the basic gates to the full macro, the layout generation is only partly automated. For a particular technology, the user needs to

provide the layout for the leaf nodes ensuring that tiling these nodes connects the common signals between them. He would also take into account the device sizes and other parameters as determined by the optimizer while drawing the layout for the leaf nodes. The layout generator simply performs a tiling operation to generate the SRAM macro with the required aspect ratio. The tiling is parameterized so that macros with different aspect ratios and different column-muxing can be easily generated. For example, Fig. 6.11 depicts the SRAM macro layout generated by ViPro for two different aspect ratios and column-muxing.

To make the SRAM layout fully automatic, the leaf nodes themselves need to be generated automatically. This could potentially be done using SKILL as well. However, SKILL doesn't automatically take into account the design rules pertaining to placement and routing of different layers and wires. Ensuring that these rules are met for any technology makes SKILL-based automatic layout of the leaf nodes challenging. Although the layout design rules vary from technology to technology, the topology of the SRAM macro remains essentially the same. Thus, abstracting away the technology-specific rules from the technology-independent topology of the SRAM will make it easier to generate the leaf node layouts automatically. This is similar to how TASE templates abstract away the technology-specific model files and device parameters from the simulation. Such a tool has recently been introduced, called PyCell Studio [77]. Using this tool, it is possible to generate the layout of a leaf node in any technology, provided a design rule lookup file for that technology. This lookup file is similar to the execution file for TASE and defines the parameters of the design rules (e.g. minimum width, spacing, area, and clearances), just as the TASE execution file defines the device parameter values and model files in the netlist used for simulation.

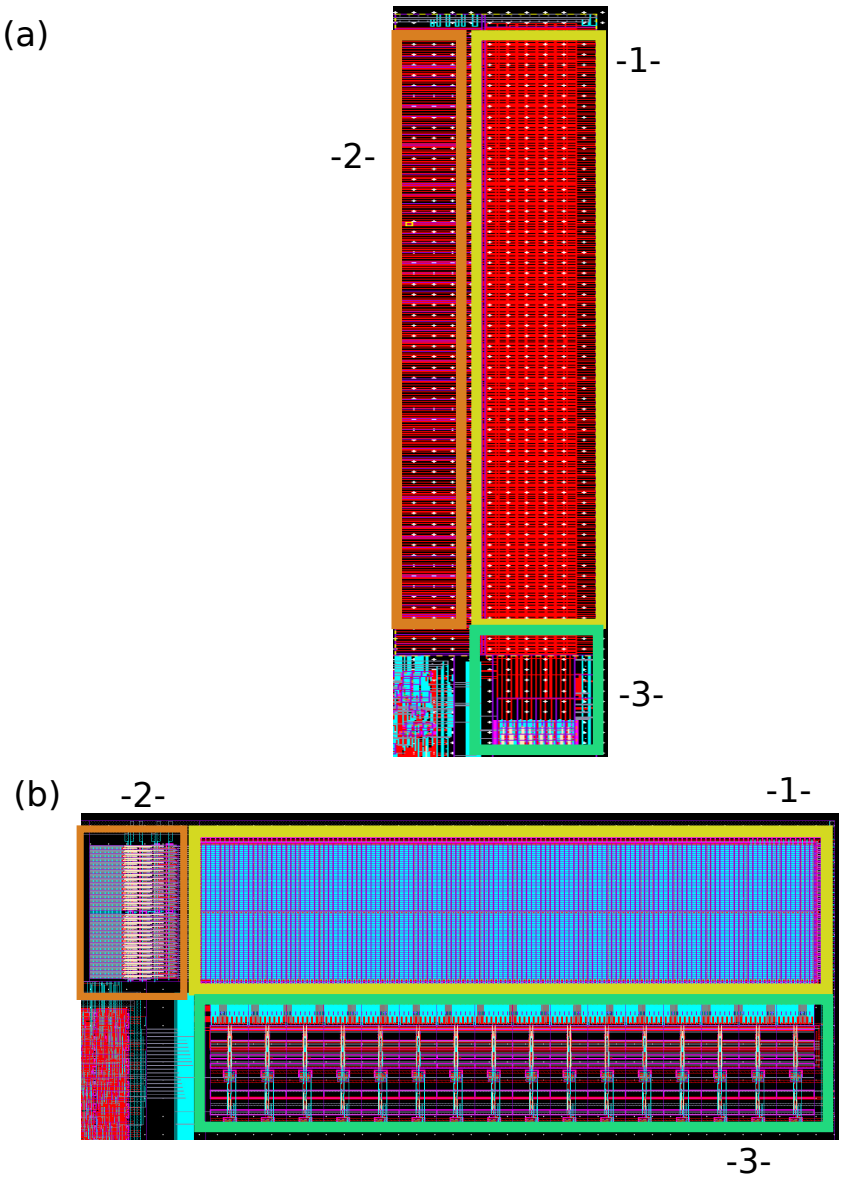


Figure 6.11: Semi-automated layout for (a) 512x16 with column-mux of 1, (b) 64x128 with column-mux of 8. The annotations 1,2, and 3 refer to the bitcell array, WL drivers, and the bitslice leaf nodes for the two macros.

6.9 Usage Examples

In this section, we demonstrate the usage of ViPro and TASE for base-case prototype generation and re-optimization for design space exploration.

6.9.1 Base-case Generation

ViPro follows the steps listed below to generate a working SRAM in a 130 nm commercial bulk CMOS technology. A key point to note here is that this technology was never used before (e.g. during development or testing of the tool) and thus can be considered as a ”new” technology.

1. Runs a preliminary characterization in TASE to estimate average gate capacitance values for use in the SRAM model calculations and sub-component characterization sweeps. PTM interconnect models encapsulated in a matlab function are also used to determine the parasitic resistance and capacitance of the metal wires needed for the SRAM model calculation (e.g. BL and WL parasitics).
2. Runs the main characterization in TASE to determine E-D characteristics of the SRAM component circuits in terms of various design knobs (e.g device sizes). As the number of knobs increases, the number of characterizations and the time taken for this step increases. However, this is a one-time step for each new technology. Changes/replacements of the component circuits for exploring the design space would require only a subset of the characterization sims to be rerun.
3. The characterization data is used by the SRAM model and the optimizer to determine the combination of knobs that result in an optimal SRAM. For the generation of the SRAM prototype in this 130 nm technology, we performed only minimal optimization. This pertained to the sizing of the buffer trees in the timing block to ensure proper generation of the conrol signals, which is needed to ensure read and write functionality. The SRAM energy and delay are not optimized for this demonstration.
4. Using the optimal buffer chain values (e.g. number of stages, fan-out) the compiler

generates the schematics of the timing block. The remainder of the schematics are sized in a fixed, worst-case manner (e.g. the precharge/write drivers are sized for the tallest column supported by the tool).

The snapshot in Fig 6.12 shows the top-level of the generated SRAM schematic. The remainder of this section demonstrates how ViPro helps quickly produce trade-off information in the form of pareto-optimal curves when exploring the design space or dealing with process or technology changes.

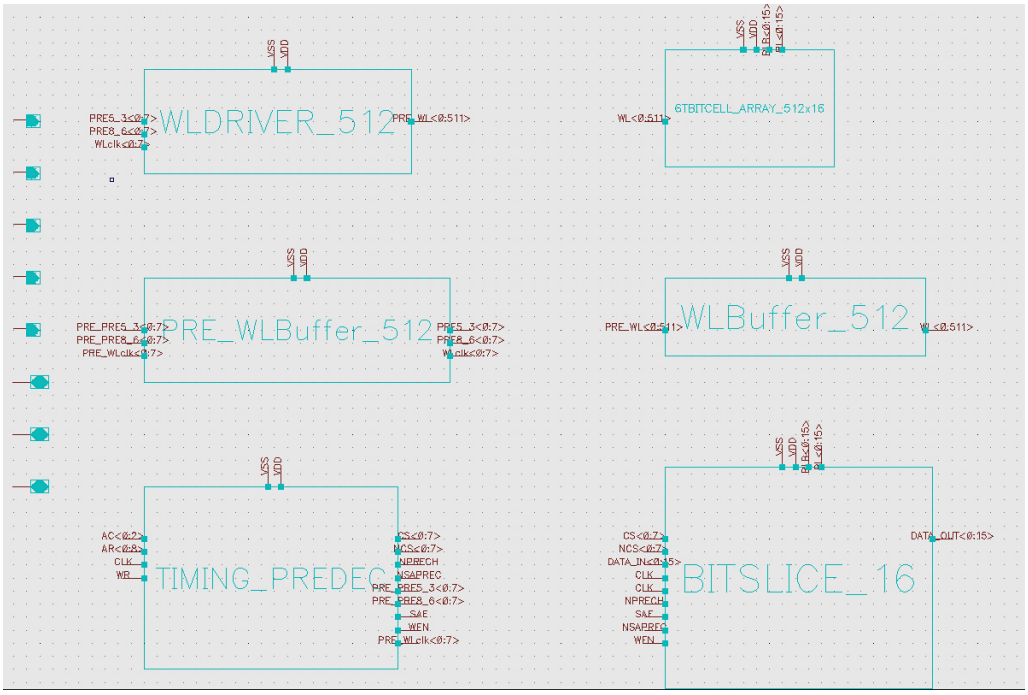


Figure 6.12: Generated SRAM schematic in the "new" 130 nm technology. The blocks starting from top-left are the WL Driver, bitcell array, WL buffer chain, bitslice, timing/predecode and predecode buffer chain.

6.9.2 Re-optimization due to Process Change

Fig. 6.13 shows the optimal E-D curve for the base-case prototype generated using 45 nm PTMs [70]. Depending on the top-level requirements of energy or delay, we can choose a design point from this curve, which provides the design parameters (rows, columns in this example) for a compiler to generate a full design. When designing SRAMs in cutting-edge technologies, designers have to deal with process models that are in constant flux. We can use ViPro to deal with this problem by rerunning the TASE characterization for using the new device models, and regenerate the base-case prototype. To demonstrate this, we increase the V_T of the transistors specified in the PTM model and re-optimize the base-case prototype for the changed model files. Now, since the new transistors are slower, fewer bit cells can be placed on a bitline to get the same performance as before. Though an experienced designer could easily draw this conclusion qualitatively, it would be much harder to decide the actual number of optimum cells per bitline, since the higher V_T transistors impact the FoMs of all the SRAM components.

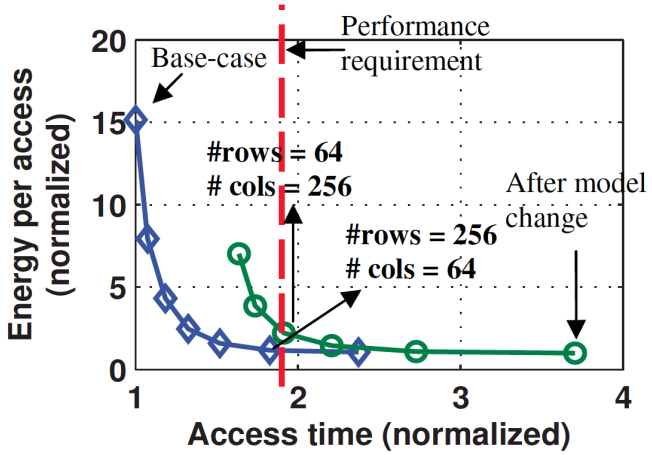


Figure 6.13: Optimal E-D curve for generated base-case prototype using 45nm PTMs and re-generated basecase E-D curve after process model change. Memory capacity = 16kb, word-size = 16 bits.

Fig. 6.13 shows the optimal E-D curve for the changed models. We observe that the same performance requirement (e.g. 1.9 units) leads to a different optimal design, with a different array configuration (e.g. 256x64 for the original models and 64x256 for the higher V_T model). Thus, ViPro provides quantitative knowledge about the impact of a process model change on the optimal SRAM design.

ViPro can similarly be used when porting over to a new technology. When porting an SRAM design to a new technology, it is likely that the optimal design changes even with no change to the design requirements. ViPro helps re-optimize the design by generating a base-case prototype in the new technology after recharacterizing the components and the technology through TASE (e.g. by defining device sizes relative to the minimum width and length), similar to the re-optimization that can be done when process models change in an existing technology. Fig. 6.14 shows the base-case tool output for 45 nm and 65 nm PTM technologies. We see that for the same energy budget of 10 units, different array configurations are optimal for different technologies (e.g. 16x1024 for 45 nm, and 32x512 for 65 nm).

6.9.3 Design Space Exploration

With the base-case prototype as a starting point, we can now explore various circuit options (e.g. different SA or bitcell) or architectural choices (e.g. different word-size), and observe how the optimal SRAM design changes with these decisions. First, suppose we want to incorporate a new SA into our SRAM design. The SA is still being designed but is targeted to have 24% lower delay at the cost of 15% higher offset, when compared to the SA in the base-case. We simply plug in these estimates for the SA component in the model of

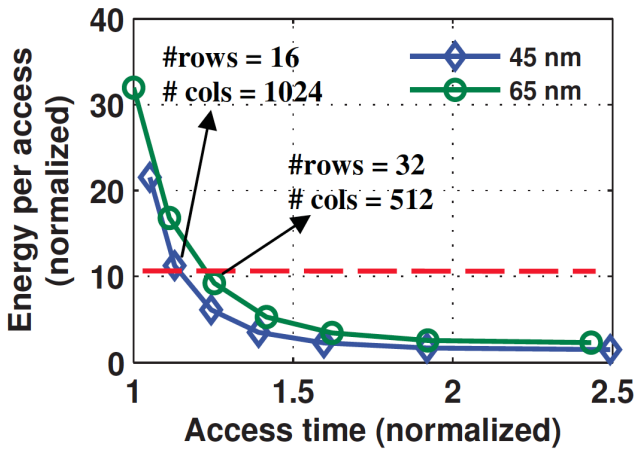


Figure 6.14: Optimal base-case E-D curves for 65nm and 45nm PTMs generated by ViPro. Memory capacity = 16 kb, word-size = 16 bits.

the SRAM (by defining them in the user input file), as we do not have a completed design of the SA yet. Now, the higher SA offset requires a larger bitline discharge. Thus, we replace the bitcell in the TASE configuration file with a larger bitcell that provides higher read current. Thus, we specify different components of the SRAM with different levels of detail. After making these changes to the SRAM, we run ViPro to reoptimize the modified design.

Fig. 6.15 shows the tool output for the base-case and the reoptimized design. We see that if the performance requirement is more than 1.6 units, the re-optimized design has higher energy than the base-case for the same delay, or conversely has higher delay than the base-case for the same energy, making the basecase more optimal. On the other hand, below 1.6 units of delay, the new design is more optimal. Thus, ViPro helps us decide whether to choose the new design or not, even when the actual SA design was not complete. Note that we have only considered energy and delay as the metrics of interest in this example. The optimization result and consequently, the design decision would change if

area is considered since the larger bitcell would increase the SRAM area.

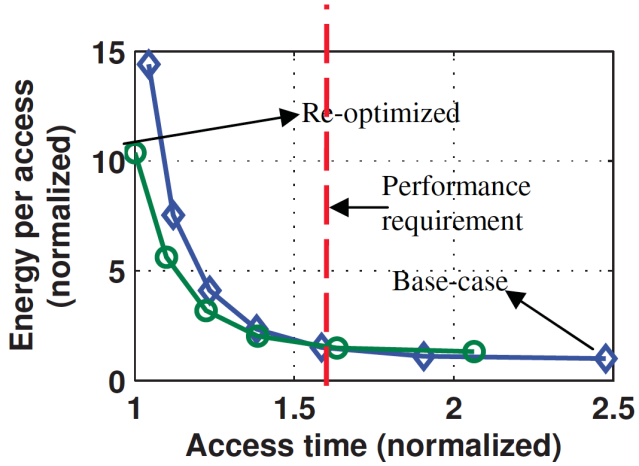


Figure 6.15: Optimal E-D curve for SRAM prototype in PTM 45nm after changes in the bitcell and SA circuits. Memory capacity = 16kb, word-size = 16 bits.

6.10 Conclusions

In this chapter we have presented two CAD tools, TASE and ViPro, that enable SRAM design automation. Both tools are technology-agnostic and work hand-in-hand to facilitate SRAM design in any new technology. While TASE acts as a platform for simulation and characterization of SRAM component circuits, ViPro helps determine the design parameters for an optimal SRAM in terms of FoMs such as energy and delay. ViPro not only provides a base-case starting point SRAM for a new technology, but can also provide FoM trade-off information (e.g. E-D curves) at every step of the design process. Finally, ViPro helps automate the schematic and layout generation for any technology.

In this dissertation, a framework has been laid down based on which TASE and ViPro can grow to become more useful, powerful tools for SRAM design. However, there is

tremendous potential for these tools to grow and become even more useful and practical. Fig. 6.16 depicts the current contributions and potential new additions to TASE and ViPro.

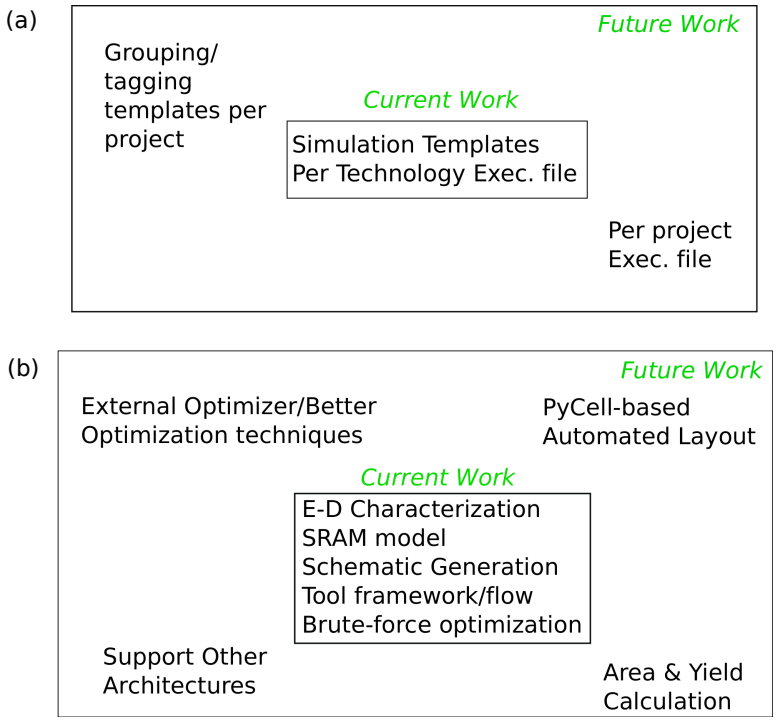


Figure 6.16: Current state of the work and potential future enhancements for (a) TASE (b) ViPro.

Chapter 7

Conclusion

7.1 Summary of Contributions

To deal with the problem of increasing power consumption in SoC systems, embedded SRAMs need to operate at a lower voltage. Since voltage scaling reduces the functional yield of the SRAM, there is a minimum acceptable voltage to which the supply votlage can be scaled (V_{MIN}). Lowering the V_{MIN} can help increase power savings while still maintaining acceptable levels of yield. In this dissertation, we have looked at alternative bitcells to lower V_{MIN} and a methodology to estimate it in a dynamic way. We have also presented productivity-enhancing SRAM design automation tools that help speed up exploring the growing SRAM design space.

7.1.1 Alternative Bitcells

1. A new 5T bitcell has been presented that uses asymmetric sizing of the cross-coupled inverters to improve RSNM.

2. Apart from improving RSNM, asymmetric sizing as a knob to trade-off critical metrics such as performance, leakage, and area has been demonstrated.
3. A detailed comparison between iso-area conventional 6T and a number of 5T bitcells with different asymmetric sizing approaches, in terms of stability, performance, and leakage has been done.
4. An analysis of the potential area savings when using the 5T has been presented.
5. The 5T has been compared with another popular alternative bitcell, the 8T, and is potentially and intermediate alternative between the 6T and the 8T in terms of stability and performance.
6. A 45 nm bulk CMOS testchip in a commercial technology has demonstrated a working 5T SRAM. The test chip also demonstrates the impact of asymmetrical sizing on bitcell writability.
7. An asymmetric 6T bitcell that remedies the drawbacks of the 5T in terms of writability has been presented.
8. A pseudo-differential sensing scheme has been demonstrated for use with single-ended bitcells such as the 5T and the asymmetric 6T.
9. A sub-45nm CMOS testchip has been fabricated to demonstrate the asymmetric 6T and single-ended sensing schemes.

7.1.2 Dynamic Write Limited V_{MIN}

1. The critical WL pulse width ($T_{WL-CRIT}$) has been presented as a measure of dynamic writability for SRAM.
2. The concepts of dynamic write noise margin and DWV_{MIN}, the dynamic write-

limited V_{MIN} have been introduced.

3. Four factors that affect the $T_{WL-CRIT}$ and DWV_{MIN} of an array have been investigated. Factors such as these cause the actual V_{MIN} of the SRAM to be different from that predicted by static noise margin metrics.
4. The impact of write assists on the DWV_{MIN} has been demonstrated. As expected, write assists lower the DWV_{MIN}. Further, the assist methods that were predicted to be the most effective in improving the static write metrics also proved to be the most effective in lowering the DWV_{MIN}.

7.1.3 SRAM Design Automation

1. A Technology Agnostic Simulation Environment (TASE) tool has been presented that enables easy porting of groups of simulations across technologies and forecasting circuit or device behavior in future techonlogies.
2. A Virtual Prototyping tool (ViPro) has been presented that enables rapid design space exploration and generation of optimal base-case SRAM prototypes in a new technology.
3. A virtual-prototype-based, semi-automated, iterative design methodology that involves the designer and leverages his expertise has been presented. ViPro produces outputs such as trade-off curves and FoM breakdown information that the designer can use to make critical design decisions even early in the design cycle.
4. A simple, simulation-based SRAM model has been presented and verified. This model forms the basis for the optimization and trade-off curve generation in ViPro.
5. Technology-agnostic schematic and layout generators based on Cadence's SKILL

language have been developed. While the schematic generator is fully automated and generates the SRAM schematic from ground up, the layout generator is semi-automated and requires manual layout of the sub-components of the SRAM.

6. A usage example has been presented that shows how ViPro can generate a working SRAM in a brand-new technology, starting from characterization of the sub-components, finding the optimal design parameters, and generating full schematic and layout of the SRAM.
7. Usage examples that demonstrate how ViPro helps re-optimize the design when component circuits or process technology changes have been presented.

7.2 Future Work

There are several additional research directions that extend the work presented in this dissertation. In particular, several new directions can be envisioned in the area of alternative bitcells and SRAM design automation. We describe the possible extensions to this dissertation in the following sub-sections.

7.2.1 Alternative Bitcells

1. The alternative bitcells proposed in this dissertation also need to be compared to state of the art bitcells such as the 6T and the 8T in terms of non-circuit aspects such as manufacturability and ligthography.
2. While we have performed extensive, bitcell-based comparisons, and a few column-level comparisons for performance, a full array-level comparison in terms of area,

power, performance, and yield is needed before the alternative bitcells proposed can truly be considered as replacements for the 6T or the 8T bitcells.

3. A more effective method of improving the write margin for the 5T can be explored. Such a method would retain the area benefit of the 5T and not sacrifice some of the improvements in cell stability, as is the case with the asymmetric 6T.
4. Measurements from the test chip taped out to demonstrate the asymmetric 6T bitcell and the pseudo-differential sensing would make this research more impactful. In particular, comparing the V_{MIN} of the asymmetric and the conventional 6T arrays would corroborate the expectation of lower V_{MIN} due to the improved noise margins.
5. Alternative power supplies were brought from off chip for the 45 nm 5T SRAM. Methods to generate these voltages on-chip can be explored.

7.2.2 Dynamic Writability

1. Since dynamic metrics are more difficult and take longer to measure, it is important to be able to predict the dynamic V_{MIN} using the static metrics alone. Thus, we can extend this work to determine and model a relationship between dynamic and static stability metrics.
2. Further investigation of factors affecting dynamic stability alone can be done. While we have looked at some of them, such as bitcell parasitics or the number of cycles prior to the first read, there are other factors such as the slew rate of the WL pulse, resistance of the vias in the bitcell etc. that could potentially impact dynamic writability as well. The better these factors are understood, the more accurately we can predict the dynamic V_{MIN} from the static metrics.

3. Another useful extension would be to model the $T_{WL-CRIT}$ distribution for a given voltage. This would help analytically predict failure probabilities without running expensive simulations, which can in turn help predict the DWV_{MIN} by finding the voltage at which the failure probability is below a certain threshold.
4. We have only looked at dynamic writability in this dissertation. While this is helpful in determining the V_{MIN} for write-limited memories, to determine the true V_{MIN} for any memory, we also need to look at dynamic read stability. While this has been explored by researchers, the concept of a read-limited V_{MIN} has not yet been developed.

7.2.3 SRAM Design Automation

1. Even more design knobs can be included in the optimization, which would make ViPro even more useful.
2. A more powerful optimization methodology can be selected to replace the exisiting brute-force search that has been used for the demonstrations in this dissertation. This would certainly be needed if more knobs are included and the brute-force search is no longer viable.
3. Currently, ViPro addresses only small memories ($<$1MB), and the architecture is a fixed, single-macro type. Figuring out how to expand the optimization to include the architecture and techniques such as ECC is needed to make the tool really useful for industry purposes.
4. Another enhancement can be at the bitcell generation level. Currently the bitcell generator only makes circuit considerations like SNM, I_{READ} etc., but without de-

vice/lithographic considerations it is hard to claim that the bitcell design is optimal. ViPro would be more attractive to industry users if the bitcell design automation was more complete and encompasses circuit and process considerations.

5. The layout generator module can be made completely automatic by using PyCell based generation of the leaf nodes of the SRAM.
6. The ideas in ViPro can be extended one level above in the design hierarchy. Instead of an SRAM, we have a system, and instead of decoders, SAs etc., we have SRAMs, Arithmetic units etc. The same kind of characterization-optimization-compilation flow used in ViPro could be extended to a system level tool as well.

7.3 Conclusion

Despite increasing variation, leakage, reliability issues, and a slew of other growing problems, CMOS technology has continued to scale way beyond the 45 nm node. This has been made possible by a combination of innovations at every level of design abstraction, ranging from new process and manufacturing improvements to new circuits and architectures to new design methodologies. SRAM has been and continues to be a technology driver and qualification vehicle for every new technology node and comprises a large area of most ICs. Thus, CMOS technology scaling is intimately linked with the ability to scale SRAM.

SRAM scaling faces even more challenges than logic design scaling due to the greater impact of variation on SRAM design, on account of the smaller bitcell geometry. While simple circuit improvements or process/manufacturing improvements alone were sufficient in older technologies to overcome these challenges, it is now becoming increasingly nec-

essary to combine innovations at multiple levels of design abstraction. The move towards high-K/metal gate or Silicon-on-Insulator, with the simulataneous application of voltage bias based read and write assist techniques or utilization of alternative bitcell structures is a prime example. Further, the explosion of the design space due to new techniques to combat scaling challenges has resulted in a productivity crisis.

In this thesis, we have started with presenting some circuit level solutions in the form of alternative 5T and asymmetric 6T bitcells to tackle the challenges facing SRAM scaling. We have also presented a methodology to determine the V_{MIN} of the SRAM more accurately using dynamic stability. Most importantly, we have presented productivity-enhancing CAD solutions in the form of ViPro and TASE that can facilitate co-design between process, circuit, and architecture by rapid generation of optimal SRAM prototypes.

Appendix A

Publications related to this dissertation

The following is the list of publications related to this thesis work.

Alternative Bitcells

1. S. Nalam and B. H. Calhoun. Asymmetric Sizing in a 45nm 5T SRAM to Improve Read Stability over 6T. In *Custom Integrated Circuit Conf. (CICC)*, pages 709-712, Sept. 2009.
2. S. Nalam, V. Chandra, C. Pietrzyk, R. Aitken, and B.H. Calhoun. Asymmetric 6T SRAM with Two-phase Write and Split Bitline Differential Sensing for Low Voltage Operation. In *IEEE Int. Symp. Quality Electronic Design (ISQED)*, pages 139-146, Mar. 2010.
3. S. Nalam and B. H. Calhoun. 5T SRAM with Asymmetric Sizing for Improved Read Stability. Accepted for publication in *IEEE J. Solid-State Circuits*.

SRAM Design Automation

4. M. Bhargava, S. Nalam, B. H. Calhoun, K. Mai. An SRAM Prototyping Tool for Rapid Sub-32nm Design Exploration and Optimization. In TECHCON, September 2009.

5. S. Nalam, M. Bhargava, K. Ringgenberg, K. Mai, and B. H. Calhoun. A Technology-Agnostic Simulation Environment (TASE) for Iterative Custom IC Design across Processes. In *IEEE Int. Conf. on Computer Design (ICCD)*, pages 523-528, Oct. 2009.
6. S. Nalam, M. Bhargava, K. Mai, and B.H. Calhoun. Virtual Prototyper (ViPro): An early design space exploration and optimization tool for SRAM designers. In *Design Automation Conf. (DAC)*, pages 138-143, Jun. 2010.
7. S. Nalam and B.H. Calhoun. TASE and ViPro: Design automation tools for nanoscale SRAM. In preparation.

Dynamic Writability for SRAM

8. J. Wang, S. Nalam and B. H. Calhoun. Analyzing Static and Dynamic Write Margin for Nanometer SRAMs. In *IEEE Int. Symp. Low Power Electronics and Design (ISLPED)*, pages 129-134, Aug. 2008.
9. S. Nalam, V. Chandra, R. C. Aitken, and B. H. Calhoun. Dynamic Write Limited Minimum Operation Voltage for Nanoscale SRAMs. In *Design, Automation and Test in Europe, Conf. and Exhibition (DATE)*, Mar. 2011.

Other SRAM work

10. R. W. Mann, S. Nalam, J. Wang, and B. H. Calhoun. Limits of Bias Based Assist Methods in Nano-Scale 6T SRAM. In *2010 IEEE Int. Symp. Quality Electronic Design*, pages 1-8, Mar. 2010.
11. J. Wang, S. Nalam, Zhenyu(Jerry) Qi, R. W. Mann, M. Stan, and B. H. Calhoun. SRAM Vmin/Yield Improvement Using Variation-Aware BTI Stress. In *Custom Integrated Circuit Conf. (CICC)*, pages 1-4, Sept. 2010.
12. R. W. Mann, J. Wang, S. Nalam, S. Khanna, G. Braceras, H. Pilo, and B. H. Calhoun. The Impact of 6T-SRAM Circuit Assist Methods on Margin and Performance. *Solid-State Electronics Journal*,54(11):1398-1407, Nov. 2010.
13. S. Khanna, S. Nalam, and B. H. Calhoun. A Fast Low-power Non-strobed Pipelined Sensing Scheme for Nano-scale SRAMs. In preparation.

Bibliography

[1] G. Moore, "The mos transistor as an individual device and in integrated arrays," *IRE International Convention Record*, vol. 13, pp. 44–52, Mar 1965.

[2] Y. Nakagome, M. Horiguchi, T. Kawahara, and K. Itoh, "Review and future prospects of low-voltage ram circuits," *IBM Journal of Research and Development*, vol. 47, no. 5.6, pp. 525 –552, sept. 2003.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Design Automation Conference, 2003. Proceedings*, june 2003, pp. 338 – 342.

[4] A. Asenov, A. Brown, J. Davies, S. Kaya, and G. Slavcheva, "Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale mosfets," *Electron Devices, IEEE Transactions on*, vol. 50, no. 9, pp. 1837 – 1852, sep. 2003.

[5] J. Wang, A. Singhee, R. Rutenbar, and B. Calhoun, "Two fast methods for estimating the minimum standby supply voltage for large srams," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 12, pp. 1908 –1920, dec. 2010.

[6] B. Zhang, A. Arapostathis, S. Nassif, and M. Orshansky, "Analytical modeling of sram dynamic stability," in *Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on*, nov. 2006, pp. 315 –322.

[7] W. Dong, P. Li, and G. Huang, "Sram dynamic stability: Theory, variability and

analysis," in *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, nov. 2008, pp. 378 –385.

[8] D. Khalil, M. Khellah, N.-S. Kim, Y. Ismail, T. Karnik, and V. De, "Accurate estimation of sram dynamic stability," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 12, pp. 1639 –1647, dec. 2008.

[9] M. Sharifkhani and M. Sachdev, "Sram cell stability: A dynamic perspective," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 2, pp. 609 –619, feb. 2009.

[10] J. Hutchby, G. Bourianoff, V. Zhirnov, and J. Brewer, "Extending the road beyond cmos," *Circuits and Devices Magazine, IEEE*, vol. 18, no. 2, pp. 28 –41, mar 2002.

[11] H. Kawasaki, M. Khater, M. Guillorn, N. Fuller, J. Chang, S. Kanakasabapathy, L. Chang, R. Muralidhar, K. Babich, Q. Yang, J. Ott, D. Klaus, E. Kratschmer, E. Sikorski, R. Miller, R. Viswanathan, Y. Zhang, J. Silverman, Q. Ouyang, A. Yagishita, M. Takayanagi, W. Haensch, and K. Ishimaru, "Demonstration of highly scaled finfet sram cells with high-kmetal gate and investigation of characteristic variability for the 32 nm node and beyond," in *Electron Devices Meeting, 2008. IEDM 2008. IEEE International*, dec. 2008, pp. 1 –4.

[12] S. Ohbayashi, M. Yabuuchi, K. Nii, Y. Tsukamoto, S. Imaoka, Y. Oda, T. Yoshihara, M. Igarashi, M. Takeuchi, H. Kawashima, Y. Yamaguchi, K. Tsukamoto, M. Inuishi, H. Makino, K. Ishibashi, and H. Shinohara, "A 65-nm soc embedded 6t-sram designed for manufacturability with read and write operation stabilizing circuits," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 4, pp. 820 –829, apr. 2007.

[13] H. Pilo, J. Barwin, G. Braceras, C. Browning, S. Burns, J. Gabric, S. Lamphier, M. Miller, A. Roberts, and F. Towler, "An sram design in 65nm and 45nm technology nodes featuring read and write-assist circuits to expand operating voltage," in *VLSI Circuits, 2006. Digest of Technical Papers. 2006 Symposium on*, 2006, pp. 15 –16.

[14] A. Bhavnagarwala, S. Kosonocky, C. Radens, Y. Chan, K. Stawiasz, U. Srinivasan, S. Kowalczyk, and M. Ziegler, "A sub-600-mv, fluctuation tolerant 65-nm cmos sram

array with dynamic cell biasing," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 4, pp. 946 –955, apr. 2008.

[15] R. W. Mann, J. Wang, S. Nalam, S. Khanna, G. Braceras, H. Pilo, and B. H. Calhoun, "Impact of circuit assist methods on margin and performance in 6t sram," *Solid-State Electronics*, vol. 54, no. 11, pp. 1398 – 1407, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/B6TY5-50GTRCY-1/2/bc05a635ff3bac857b837b42828dd8bc

[16] L. Chang, Y. Nakamura, R. Montoye, J. Sawada, A. Martin, K. Kinoshita, F. Gebara, K. Agarwal, D. Acharyya, W. Haensch, K. Hosokawa, and D. Jamsek, "A 5.3ghz 8t-sram with operation down to 0.41v in 65nm cmos," in *VLSI Circuits, 2007 IEEE Symposium on*, jun. 2007, pp. 252 –253.

[17] B. Calhoun and A. Chandrakasan, "A 256kb sub-threshold sram in 65nm cmos," in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, feb. 2006, pp. 2592 –2601.

[18] T.-H. Kim, J. Liu, J. Keane, and C. Kim, "A high-density subthreshold sram with data-independent bitline leakage and virtual ground replica scheme," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, feb. 2007, pp. 330 –606.

[19] A. Bhavnagarwala, X. Tang, and J. Meindl, "The impact of intrinsic device fluctuations on cmos sram cell stability," *Solid-State Circuits, IEEE Journal of*, vol. 36, no. 4, pp. 658 –665, apr. 2001.

[20] R. Kumar and G. Hinton, "A family of 45nm ia processors," in *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, feb. 2009, pp. 58 –59.

[21] D. Wendel, R. Kalla, R. Cargoni, J. Clables, J. Friedrich, R. Frech, J. Kahle, B. Sinharoy, W. Starke, S. Taylor, S. Weitzel, S. Chu, S. Islam, and V. Zyuban, "The implementation of power7tm: A highly parallel and scalable multi-core high-end server

processor," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, feb. 2010, pp. 102 –103.

[22] R. Jotwani, S. Sundaram, S. Kosonocky, A. Schaefer, V. Andrade, G. Constant, A. Novak, and S. Naffziger, "An x86-64 core implemented in 32nm soi cmos," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, feb. 2010, pp. 106 –107.

[23] J. Pille, D. Wendel, O. Wagner, R. Sautter, W. Penth, T. Froehnel, S. Buettner, O. Torreiter, M. Eckert, J. Paredes, D. Hrusecky, D. Ray, and M. Canada, "A 32kb 2r/1w l1 data cache in 45nm soi technology for the power7tm processor," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, feb. 2010, pp. 344 –345.

[24] M. Yamaoka, N. Maeda, Y. Shinozaki, Y. Shimazaki, K. Nii, S. Shimada, K. Yanagisawa, and T. Kawahara, "Low-power embedded sram modules with expanded margins for writing," in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, feb. 2005, pp. 480 –611 Vol. 1.

[25] Y. Morita, H. Fujiwara, H. Noguchi, Y. Iguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "An area-conscious low-voltage-oriented 8t-sram design under dvs environment," in *VLSI Circuits, 2007 IEEE Symposium on*, jun. 2007, pp. 256 –257.

[26] I. Carlson, S. Andersson, S. Natarajan, and A. Alvandpour, "A high density, low leakage, 5t sram for embedded caches," in *Solid-State Circuits Conference, 2004. ESSCIRC 2004. Proceeding of the 30th European*, sep. 2004, pp. 215 – 218.

[27] H. Tran, "Demonstration of 5t sram and 6t dual-port ram cell arrays," in *VLSI Circuits, 1996. Digest of Technical Papers., 1996 Symposium on*, jun. 1996, pp. 68 –69.

[28] M. Wieckowski, S. Patil, and M. Margala, "Portless sram-a high-performance alternative to the 6t methodology," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 11, pp. 2600 –2610, nov. 2007.

[29] S. Nalam and B. Calhoun, "Asymmetric sizing in a 45nm 5t sram to improve read stability over 6t," in *Custom Integrated Circuits Conference, 2009. CICC '09. IEEE*, 2009, pp. 709–712.

[30] E. Seevinck, F. List, and J. Lohstroh, "Static-noise margin analysis of mos sram cells," *Solid-State Circuits, IEEE Journal of*, vol. 22, no. 5, pp. 748 – 754, oct. 1987.

[31] A. Srivastava, "Simultaneous vt selection and assignment for leakage optimization," in *Low Power Electronics and Design, 2003. ISLPED '03. Proceedings of the 2003 International Symposium on*, aug. 2003, pp. 146 – 151.

[32] S. Nalam, V. Chandra, C. Pietrzyk, R. Aitken, and B. Calhoun, "Asymmetric 6t sram with two-phase write and split bitline differential sensing for low voltage operation," in *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, mar. 2010, pp. 139 –146.

[33] M. Khellah, N. S. Kim, Y. Ye, D. Somasekhar, T. Karnik, N. Borkar, G. Pandya, F. Hamzaoglu, T. Coan, Y. Wang, K. Zhang, C. Webb, and V. De, "Process, temperature, and supply-noise tolerant 45 nm dense cache arrays with diffusion-notch-free (dnf) 6t sram cells and dynamic multi-vcc circuits," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 4, pp. 1199 –1208, apr. 2009.

[34] Y. Morita, H. Fujiwara, H. Noguchi, K. Kawakami, J. Miyakoshi, S. Mikami, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A vth-variation-tolerant sram with 0.3-v minimum operation voltage for memory-rich soc under dvs environment," in *VLSI Circuits, 2006. Digest of Technical Papers. 2006 Symposium on*, 2006, pp. 13 –14.

[35] B. Calhoun and A. Chandrakasan, "Static noise margin variation for sub-threshold sram in 65-nm cmos," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 7, pp. 1673 –1679, jul. 2006.

[36] Y. Ye, M. Khellah, D. Somasekhar, and V. De, "Evaluation of differential vs. single-ended sensing and asymmetric cells in 90 nm logic technology for on-chip caches," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, pp. 4 pp. –966.

[37] N. Gierczynski, B. Borot, N. Planes, and H. Brut, "A new combined methodology for write-margin extraction of advanced sram," in *Microelectronic Test Structures, 2007. ICMTS '07. IEEE International Conference on*, mar. 2007, pp. 97 –100.

[38] N. Verma and A. Chandrakasan, "A high-density 45nm sram using small-signal non-strobed regenerative sensing," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, feb. 2008, pp. 380 –621.

[39] A. Kawasumi, T. Yabe, Y. Takeyama, O. Hirabayashi, K. Kushida, A. Tohata, T. Sasaki, A. Katayama, G. Fukano, Y. Fujimura, and N. Otsuka, "A single-power-supply 0.7v 1ghz 45nm sram with an asymmetrical unit--ratio memory cell," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, feb. 2008, pp. 382 –622.

[40] W. Dehaene, S. Cosemans, A. Vignon, F. Catthoor, and P. Geens, "Embedded sram design in deep deep submicron technologies," in *Solid State Circuits Conference, 2007. ESSCIRC 2007. 33rd European*, sep. 2007, pp. 384 –391.

[41] A. Moshovos, B. Falsafi, F. Najm, and N. Azizi, "A case for asymmetric-cell cache memories," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 7, pp. 877 – 881, jul. 2005.

[42] N. Tzartzanis and W. Walker, "A differential current-mode sensing method for high-noise-immunity, single-ended register files," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, 2004, pp. 506 – 543 Vol.1.

[43] J. Poulton, "An embedded dram for cmos asics," in *Advanced Research in VLSI, 1997. Proceedings., Seventeenth Conference on*, sep. 1997, pp. 288 –302.

[44] J. Yeung and H. Mahmoodi, "Robust sense amplifier design under random dopant fluctuations in nano-scale cmos technologies," in *SOC Conference, 2006 IEEE International*, sep. 2006, pp. 261 –264.

[45] J. Wang, A. Singhee, R. Rutenbar, and B. Calhoun, "Statistical modeling for the minimum standby supply voltage of a full sram array," in *Solid State Circuits Conference, 2007. ESSCIRC 2007. 33rd European*, sep. 2007, pp. 400 –403.

[46] K. Agarwal and S. Nassif, "Statistical analysis of sram cell stability," in *Design Automation Conference, 2006 43rd ACM/IEEE*, 2006, pp. 57 –62.

[47] A. Bhavnagarwala, S. Kosonocky, C. Radens, K. Stawiasz, R. Mann, Q. Ye, and K. Chin, "Fluctuation limits and scaling opportunities for cmos sram cells," in *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*, dec. 2005, pp. 659 –662.

[48] J. Wang, S. Nalam, and B. H. Calhoun, "Analyzing static and dynamic write margin for nanometer srams," in *ISLPED '08: Proceeding of the 13th international symposium on Low power electronics and design*. New York, NY, USA: ACM, 2008, pp. 129–134. [Online]. Available: http://portal.acm.org/ft_gateway.cfm?id=1393954&type=external&coll=Portal&dl=GUIDE&CFID=105953287&CFTOKEN=22466367

[49] S. O. Toh, Z. Guo, and B. Nikolic, "Dynamic sram stability characterization in 45nm cmos," in *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*, jun. 2010, pp. 35 –36.

[50] O. Hirabayashi, A. Kawasumi, A. Suzuki, Y. Takeyama, K. Kushida, T. Sasaki, A. Katayama, G. Fukano, Y. Fujimura, T. Nakazato, Y. Shizuki, N. Kushiyama, and T. Yabe, "A process-variation-tolerant dual-power-supply sram with $0.179\mu^2$ cell in 40nm cmos using level-programmable wordline driver," in *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, feb. 2009, pp. 458 –459,459a.

[51] H. Yang, R. Wong, R. Hasumi, Y. Gao, N. Kim, D. Lee, S. Badrudduza, D. Nair, M. Ostermayr, H. Kang, H. Zhuang, J. Li, L. Kang, X. Chen, A. Thean, F. Arnaud, L. Zhuang, C. Schiller, D. Sun, Y. Teh, J. Wallner, Y. Takasu, K. Stein, S. Samavedam, D. Jaeger, C. Baiocco, M. Sherony, M. Khare, C. Lage, J. Pape, J. Sudijono, A. Steegen, and S. Stiffler, "Scaling of 32nm low power sram with high-k metal gate," in

Electron Devices Meeting, 2008. IEDM 2008. IEEE International, dec. 2008, pp. 1 –4.

[52] N. Shibata, H. Kiya, S. Kurita, H. Okamoto, M. Tan'no, and T. Douseki, "A 0.5-v 25-mhz 1-mw 256-kb mtcmos/soi sram for solar-power-operated portable personal digital equipment - sure write operation by using step-down negatively overdriven bitline scheme," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 3, pp. 728 – 742, mar. 2006.

[53] V. Chandra, C. Pietrzyk, and R. Aitken, "On the efficacy of write-assist techniques in low voltage nanoscale srams," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, mar. 2010, pp. 345 –350.

[54] G. Huang, W. Dong, Y. Ho, and P. Li, "Tracing sram separatrix for dynamic noise margin analysis under device mismatch," in *Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International*, sep. 2007, pp. 6 –10.

[55] B. Amrutur and M. Horowitz, "A replica technique for wordline and sense control in low-power sram's," *Solid-State Circuits, IEEE Journal of*, vol. 33, no. 8, pp. 1208 –1219, aug. 1998.

[56] K. Takeda, T. Saito, S. Asayama, Y. Aimoto, H. Kobatake, S. Ito, T. Takahashi, K. Takeuchi, M. Nomura, and Y. Hayashi, "Multi-step word-line control technology in hierarchical cell architecture for scaled-down high-density srams," in *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*, jun. 2010, pp. 101 –102.

[57] Y. Chung and S.-H. Song, "Implementation of low-voltage static ram with enhanced data stability and circuit speed," *Microelectron. J.*, vol. 40, no. 6, pp. 944–951, 2009.

[58] T. Suzuki, H. Yamauchi, Y. Yamagami, K. Satomi, and H. Akamatsu, "A stable 2-port sram cell design against simultaneously read/write-disturbed accesses," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 9, pp. 2109 –2119, sep. 2008.

[59] M. Bhargava, M. McCartney, A. Hoefler, and K. Mai, "Low-overhead, digital offset compensated, sram sense amplifiers," in *Custom Integrated Circuits Conference, 2009. CICC '09. IEEE*, sep. 2009, pp. 705 –708.

[60] Y. B. Kim, Y.-B. Kim, and F. Lombardi, "New sram cell design for low power and high reliability using 32 nm independent gate finfet technology," in *Design and Test of Nano Devices, Circuits and Systems, 2008 IEEE International Workshop on*, sep. 2008, pp. 25 –28.

[61] P. Shivakumar and N. P. Jouppi, "Cacti 3.0: An integrated cache timing, power, and area model," Western Research Laboratory, Tech. Rep., 2002.

[62] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing nuca organizations and wiring alternatives for large caches with cacti 6.0," in *MICRO 40: Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 3–14.

[63] ITRS, "International technology roadmap for semiconductors," 2006. [Online]. Available: http://www.itrs.net/

[64] X. Bai, C. Visweswariah, P. Strenski, and D. Hathaway, "Uncertainty-aware circuit optimization," in *Design Automation Conference, 2002. Proceedings. 39th*, 2002, pp. 58 – 63.

[65] V. Sundararajan, S. Sapatnekar, and K. Parhi, "Fast and exact transistor sizing based on iterative relaxation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 21, no. 5, pp. 568 –581, may. 2002.

[66] K. Chakraborty, S. Kulkami, M. Bhattacharya, P. Mazumder, and A. Gupta, "A physical design tool for built-in self-repairable rams," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 352 –364, apr. 2001.

[67] A. Chandna, "Gaas mesfet sram design for embedded applications," Ph.D. dissertation, University of Michigan, 1995.

[68] R. Sredojevic and V. Stojanovic, "Optimization-based framework for simultaneous circuit-and-system design-space exploration: A high-speed link example," in *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, nov. 2008, pp. 314 –321.

[69] S. Nalam, M. Bhargava, B. H. Calhoun, and K. Mai, "Sram circuit design and optimization at 45nm and below," C2S2 Annual Review Poster, apr. 2009.

[70] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *Electron Devices, IEEE Transactions on*, vol. 53, no. 11, pp. 2816 –2823, nov. 2006.

[71] S. Nalam, M. Bhargava, K. Ringgenberg, K. Mai, and B. Calhoun, "A technology-agnostic simulation environment (tase) for iterative custom ic design across processes," in *Computer Design, 2009. ICCD 2009. IEEE International Conference on*, oct. 2009, pp. 523 –528.

[72] S. Boyd, S. Kim, D. Patil, and M. Horowitz, "Digital circuit optimization via geometric programming," in *Operations Research*, vol. 53, 2005, pp. 899 – 932.

[73] J. Lavaei, A. Babakhani, A. Hajimiri, and J. Doyle, "Solving large-scale linear circuit problems via convex optimization," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, dec. 2009, pp. 4977 –4984.

[74] MOSEK, "The mosek optimization software," Copenhagen, Denmark. [Online]. Available: http://www.mosek.com/

[75] D. Patil and S. Kim, "Stanford circuit optimization tool (scot) user guide," Stanford University, Tech. Rep., 2007.

[76] S. Huntzicker, M. Dayringer, J. Soprano, A. Weerasinghe, D. Harris, and D. Patil, "Energy-delay tradeoffs in 32-bit static shifter designs," in *Computer Design, 2008. ICCD 2008. IEEE International Conference on*, oct. 2008, pp. 626 –632.

[77] Ciranova, "Pycell studio," Santa Clara. [Online]. Available: http://www.ciranova.com/products/pycell_studio.php